2001-12

Network application server using Extensible Mark-up Language (XML) to support distributed databases and 3D environments

Williams, Clifton James

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/4047

NAVAL POSTGRADUATE SCHOOL
Monterey California USA

THESIS

NETWORK APPLICATION SERVER USING EXTENSIBLE MARK-UP LANGUAGE (XML) TO SUPPORT DISTRIBUTED DATABASES AND 3D ENVIRONMENTS

by

Clifton James Williams

December 2001

Thesis Advisor:	Don Brutzman
Thesis Advisor:	Daniel Dolk

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY	**2. REPORT DATE** December 2001	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE: Network Application Server Using Extensible Mark-Up Language (XML) To Support Distributed Databases And 3D environments.	**5. FUNDING NUMBERS**
6. AUTHOR Williams, Clifton James	
7. PERFORMING ORGANIZATION NAME (S) AND ADDRESS (ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME (S) AND ADDRESS (ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE**

13. ABSTRACT

This thesis contributes to the US Navy forces maintaining information superiority in a Network Centric Warfare environment. This research develops an Extensible Markup Language XML-based Web Publishing Framework, which supports the Web Enabled Navy (WEN) architecture infrastructure. The Web application framework easily supports connections to multiple distributed databases and XML-based presentations, specifically three-dimensional (3D) simulations utilizing Extensible 3D (X3D) and Virtual Reality Modeling Language (VRML). This research evaluates and demonstrates the functionality of an XML-based Web Publishing Framework. The research demonstrates ability to connect to both XML and non-XML heterogeneous database systems and provides a framework for the distribution data across heterogeneous systems. The system supports usage in multi-tier network architecture. 3D modeling and simulations provide insights into operations that cannot be realized using standard two-dimensional (2D) renditions.

The development of an application server to support 3D modeling and simulations for operational planning will provide the U.S. Navy a better way of realizing operational limits. This research addresses the Web publishing framework, which supports the development XML based data exchange, and the development of an "on the fly" X3D simulation presentation. This research evaluated the functionality of an XML based Web framework, which allows for the separation between application programming and Web presentation. This incorporation of a Model, View, and Controller (MVC) design approach provides a clean separation between different components (logic, presentation, and data) of information and the programming functionality. The Web framework addressed ability to provide a framework for the distribution data across heterogeneous systems.

14. SUBJECT TERMS Command, Control, and Communications, Computing and Software Extensible Mark-up Language (XML), Java Servlets, Web Application Framework, Distributed Databases, Heterogeneous Systems, Hypertext Markup Language (HTML), Virtual Reality Modeling Language (VRML), Extensible 3D (X3D), and Extensible Stylesheet Language (XSL)	**15. NUMBER OF PAGES**
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

NETWORK APPLICATION SERVER USING EXTENSIBLE MARK-UP LANGUAGE (XML) TO SUPPORT DISTRIBUTED DATABASES AND 3D ENVIRONMENTS

Clifton J. Williams
Lieutenant, United States Navy
B.E.E., Auburn University, 1993

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

and

MASTER OF SCIENCE IN INFORMATION SYSTEMS TECHNOLOGY

from the

NAVAL POSTGRADUATE SCHOOL

December 2001

Author: ______________________________
Clifton J. Williams

Approved by: ______________________________
Don Brutzman, Thesis Advisor

Daniel R. Dolk
____
Daniel R. Dolk, Thesis Advisor

____
LCDR Chris Eagle, Chair
Computer Science Department

____
George Connor, Chair
Information Systems Academic Group

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Advances in computer communication technology and an increased awareness of how enhanced information access can lead to improved capabilities, are the driving forces behind the interest in the integration of current distributed systems and stand-alone systems. However, differences in hardware, software architectures, operating systems, host languages, and data representation have resulted in scores of stovepipe system architectures that are unable to interoperate properly.

This thesis contributes to the US Navy forces' information superiority in a Network Centric Warfare environment. This research develops an Extensible Markup Language (XML) based Web application framework, which supports the Web Enabled Navy (WEN) architecture infrastructure. The Web application framework easily supports connections to multiple distribute databases and XML-based presentations specifically three-dimensional (3D) simulations utilizing Extensible 3D (X3D) and Virtual Reality Modeling Language (VRML).

This research evaluates and demonstrates the functionality of an XML-based Web framework. The framework allows for the separation between application programming and Web presentation. The research demonstrates ability to connect to both XML and non-XML heterogeneous database systems and provides a framework for the distribution data across heterogeneous systems. The system supports usage in multi-tier network architecture. 3D modeling and simulations provide insights into operations that are unrealized using standard two-dimensional (2D) renditions. Today's planning and modeling systems use 2D representations of the 3D battle-space. This presents a challenge for planners, commanders, and operators to understand the true nature of the battle-space with regards to the easiest routes to take, places to take cover, obstacles which may impede traffic, and places where the enemy may be hiding, to name a few.

The development of an application server to support 3D modeling and simulations for operational planning will provide the U.S. Navy a better way of realizing operational limits. This research addresses the Web application framework, which supports the development XML-based data exchange, and the development of an "on the fly" X3D simulation presentation.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF SYMBOLS, ACRONYMS, AND/OR ABBREVIATIONS

2D	Two-Dimensional
3D	Three-Dimensional
API	Application Program Interface
ASP	Active Server Pages
COE	Common Operating Environment
CSS	Cascading Style Sheet
CGI	Common Gateway Interface
CPU	Central Processing Unit
DAP	Directory Access Protocol
DBA	Database Administrator
DBMS	Database Management System
DHTML	Dynamic Hyper Text Markup Language
DII	Defense Information Infrastructure
DISA	Defense Information Systems Agency
DoD	Department of Defense
DOM	Document Object Model
DMSO	Defense Modeling and Simulation Office
DTD	Data Type Definition
EJB	Enterprise Java Beans
ESQL	Extensible Standard Query Language Tag Library
FO	Formatted Object
FOP	Formatted Objects Processor
FTP	File Transfer Protocol
GNP	Gross National Product
GUI	Graphical User Interface
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
IDE	Integrated Development Environment
IIS	Internet Information Server
J2EE	Java 2 Enterprise Edition
JAXP	Java API for XML Processing
JDBC	Java Database Connection
JNDI	JAVA Naming and Directory Interfaces
JSP	Java Server Pages
JVM	Java Virtual Machine
LDAP	Lightweight Directory Access Protocol
MGRS	Military Grid Reference System
MVC	Model, View, Controller
NCSA	National Center for Supercomputing Applications
NXI	Navy XML Infrastructure
OOP	Object-Oriented Programming
ODBC	Object Database Connection

PDA	Personal Digital Assistant
PDF	Portable Document Format
RDBMS	Relational Database Management Systems
RMI	Remote Method Invocation
ROE	Rules of Engagement
SAX	Simple API for XML
SQL	Standard Query Language
SRS	Software Requirements Specification
SSL	Secure Socket Layer
SVG	Scalable Vector Graphics
TCP/IP	Transmission Control Protocol/ Internet Protocol
TBD	To Be Determined
URI	Uniform Resource Indicator
URL	Uniform Resource Locator
USMC	United States Marine Corps
USMTF	United States Message Text Forma
USN	United States Navy
VRML	Virtual Reality Modeling Language
W3C	World Wide Web Consortium
WAN	Wide Area Network
WAP	Wireless Access Protocol
WEN	Web Enabled Navy
WML	Wireless Markup Language
WWW	World Wide Web
X3D	Extensible 3D Markup Language
XHTML	Extensible Hypertext Markup Language
Xlink	XML Link References
XML	Extensible Mark-up Language
XML-FO	Extensible Markup Language Formatting Object
XML-MTF	Extensible Markup Language Message Text Format
Xpath	XML Path Language
XPointer	XML Pointer References
XSL	Extensible Stylesheet Language
XSL-FO	XSL Formatting Objects
XSLNS	XSL Name Space
XSLT	XSL Transformations
XSP	Extensible Server Pages
XSSI	Extended server side includes

ACKNOWLEDGMENTS

First, I would thank God for answering my prayers and truly blessing me here at Monterey with the opportunity to advance my education in the computer industry. All my knowledge and wisdom flows from him. Second, I thank my beautiful wife and best friend Nancy, my triplet sons Colin, Ryan and Nicholas and son Clifton for making life truly worth living. Lastly, I owe a debt of gratitude to my fellow students and instructors who have made my time here at NPS a very pleasant and memorable experience.

-- Clifton J. Williams

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. PROBLEM STATEMENT

Extensible Mark-up Language (XML) based Web application framework provides a system for the distribution of data across heterogeneous systems. An XML-based Web framework allows for the separation between application programming and Web presentation. This type of application server supports connections between both XML and non-XML heterogeneous database systems. The Web frame supports hypertext markup language (HTML) as well as XML. An XML-based Web Publishing Frameworkprovides a framework for distributing and visualizing three-dimensional simulation utilizing Extensible 3D (X3D) and Virtual Reality Modeling Language (VRML) over a wide area network (WAN). This framework supports the use of multiple Extensible Stylesheet Language (XSL) Stylesheets, to enable translations between XML information and VRML three-dimensional objects.

The purpose of this thesis is to support information superiority in a Network Centric environment and support the Web Enabled Navy (WEN) Architecture infrastructure. The objective of this research is to demonstrate and evaluate the functionality of an XML-based Web framework, which allows for the separation between application programming and Web presentation. The Web framework research will demonstrate ability to connect to both XML and non-XML heterogeneous databases and provide a framework for distributing data across heterogeneous systems. The Web framework supports the Navy Extensible Mark-up Language Infrastructure (NXI). The XML based Web application framework provides a framework for distributing and visualizing three-dimensional simulation utilizing X3D and VRML. Specifically, this thesis illustrates the development of an XML-based Application Server that supports the use of multiple XSL stylesheets to enable translations between XML information and VRML three-dimensional object representations.

B. BACKGROUND

Current littoral operations doctrine still uses the command and control systems established to handle the Russian threat in a blue-water, open ocean environment. In the past, when the US Naval Force deployed, the information exchange between the Battle group units came via hardcopy naval message, tactical data links, and voice and data radio circuits. Battle group units were responsible for various warfare areas in a decentralized command and control structure governed by the commander's intent and rules of engagement (ROE). Sensors and weapons belonged to individual battle group units and each unit was responsible for their safe and effective employment in accordance with the commander's intent. Special instructions, supplemental ROE, and pre-planned responses in accordance with battle group operating orders were in place to handle rapid changes in the tactical situations.

The Navy and Marine Corps of 2020 will more frequently operate in the littoral, conducting the core missions of Undersea Warfare, Air Warfare, Surface Warfare, and Land Attack. A networked infrastructure that will leverage battle-group firepower and sensor capabilities will need to support these rapidly changing littoral environment operations. Sensors and weapons will no longer be "stove piped" to one unit. To conduct operations in this manner, new types of operation simulation planning and battle-space awareness will be needed that enable a more robust flow of information between the units in a manner that ensures timeliness, and is sufficient to handle the added complexity of the littoral environment.

The US Navy communications infrastructure must support the deployment of forces for operations. Internet type Transmission Control Protocol/Internet Protocol (TCP/IP) information exchange systems will require transferring data between senders and receivers. "Information superiority will enhance the capability of joint force commander to understand the situation, determine the effects desired, select a course of action and the forces to execute it, accurately assess the effects of the action, and reengage as necessary while minimizing collateral damage." [Joint Vision 2020, PG 22] This research will contribute to the US Navy forces' information superiority in a Network

Centric Warfare environment. This research develops an XML-based Web application framework, which supports the WEN architecture infrastructure. [WEN01] The proposed Web application framework easily supports distribute databases and XML-based presentations specifically three-dimensional simulations utilizing X3D and VRML. The system supports incorporation into a multi-tier network architecture to link various organic and inorganic actors in a virtual Combat Information Center. Three-dimensional (3D) modeling and simulations provide insights into operations, which cannot be unrealized using standard two-dimensional (2D) renditions. The development of an application server to support three-dimensional modeling and simulations for operational planning will provide the U.S. Navy a better way of realizing operational limits. The research seeks to addresses the Web application framework which supports the development XML-based data exchange and the development of an on fly X3D simulation presentations.

C. PROBLEM

The principal problem to be resolved is how to share information between two or more systems using the current Web framework. Ancillary questions are how to extend current relational database management systems (RDBMS) to support the current Web technology using large data storage systems, how to enable application servers to quickly change information representations in support of 3D simulations, how to achieve these objectives in an environment consisting of heterogeneous databases across heterogeneous systems. An XML-based Web framework application server provides the solution to these issues.

This research demonstrates and evaluates the functionality of an XML based Web framework, which allows for the separation between application programming and Web presentation. The Web framework research demonstrates ability to connect to both XML and non-XML heterogeneous database systems and provides a framework for the distribution data across heterogeneous systems. The XML-based Web Publishing Frameworkprovides a framework for distributing and visualizing three-dimensional simulation utilizing X3D and VRML.

D. ROOT CAUSE

The Defense Information Infrastructure (DII) Master Plan states "Information Technology infrastructure is composed of many disparate underlying computing configurations, designed and implemented at different times to meet different requirements." Given the varying perceived service-specific requirements of DoD users and the distributed execution authority within DoD, it is reasonable to assume that heterogeneity of our systems will continue to be prevalent [DII98].

Heterogeneity is inevitable and is a phenomenon we must handle effectively to deal with if we are to control costs and realize the full potential of our investments. Increased costs are due to two factors. First, many of these systems unintentionally produce similar or complementary information. Second, advances in computer communications and networking technologies, a shrinking defense budget, and an increase in joint and combined operations now make it possible and necessary to integrate. Funding is not available to procure new integrated network-centric systems to replace the thousands of DoD legacy systems currently operating and produced under the stand-alone / closed-system paradigm.

The proliferation of these non-interoperable legacy systems has imposed delays, costs, and friction in nearly every area of the military. By conservative estimates, about 2 percent of the gross national product (GNP) can be attributed to inefficiencies such as redundant information entry, data conversion and system incompatibilities [SB01]. The United States, having a GNP greater than 9.8 trillion dollars in 2000 [OECD01], wasted at least 197 billion dollars on these inefficiencies. To put this in perspective, these inefficiencies amount to more than double the total same year budget of the Navy and Marine Corps combined [DL99].

The cost of not integrating is overwhelmingly evident. As the defense budget shrinks, these inefficiencies will take a larger slice of the total and result in less funding needed to strengthen our war fighting capabilities. DoD must move to meet the computing goal of evolved ability and interoperability within this heterogeneous environment in the most intelligent and cost efficient manner.

The current method for integration of legacy systems involves a careful, time-intensive study of a producer system's exported information. The system integrator then determines if this information is of value to a consumer system and what formatting or transformations are required. After mapping the exported data type to the imported data type, the system integrator then writes a software "wrapper" which will perform a translation from the producer's representation to the consumer's. Unfortunately, this cycle repeats for every new piece of information a consumer or producer would like to make available to the system. Currently there is little or no computer assistance for performing this task. [Lytt00]

The approach proposed by this thesis is to use an XML Web application server as a more efficient means of providing systems interoperability and then to develop XSL stylesheets to enable automated translations between XML and users' desired system outputs. This approach will also allow the DoD to lower system integrations costs. An XML-based Web framework will allow the isolation of internal code of legacy systems with an XML "wrapper". The framework will view each system input as a black box and deal only with the interfaces. This system allows for heterogeneous distributed database connections, which support XML and non-XML systems without modifying the DBMS structure.

The benefits of using an XML-based application server to move towards interoperability include:

- Reduction in Web site development and maintenance costs, thus freeing monies and providing more flexibility in making funding choices for other programs.
- Incorporation of a Model, View, Controller (MVC) design approach. MVC approach to design provides a clean separation between different components (logic, presentation, and data) of information and the programming functionality.

- More timely and efficient integration of arbitrary legacy systems and data sources into an existing network.
- Ability to easily add new systems to a network without impacting existing systems in the network.
- A framework for distributing and 3D simulations utilizing X3D and VRML.
- A revolutionary and synergistic effect on how our systems and warfighters communicate and share information.

E. METHODOLOGY

This research start with a thorough review of various established Web frameworks, application servers, XML parsing methods, and tools needed to help in the integration of XML in an application server processes. This establishes the background for the selection of the framework adopted in this thesis. The next phase of this research examines the current XML representation of the VRML in the context of 3D simulations and suggests modifications to support a Web application framework. This leads to a specification of requirements which form the foundation for building a functional prototype of the XML Web application framework to support Web based XML operations. The final step is evaluations of the Web framework prototype performance against traditional non-XML-Based Web site performance using Java Servlet technology with out XML.

F. SCOPE

Because the XML-based Web Publishing Frameworkcan support an unlimited amount of Web applications services, all of the constraints below are imposed to limit the scope of this thesis to demonstrating an effective proof of principle:

- Extensible Markup Language (XML) is used as the interoperability language for data integration. Chapter 3 is will cover the merits of XML.

- Open source software development is used exclusively in order to reduce development costs of the Web framework.
- The Web-publishing framework uses Java Technology and supports the Java 2 Enterprise Edition (J2EE).
- Java is not only the best language for integration with XML, but it is also the foundation for most of the open-source XML, Servlets, editors, parsers, validators, and tools. This wealth of open-source material significantly expedited the building of the prototype.

G. CHAPTER OUTLINE

This chapter analyzes the problems and discusses the motivation behind the effort, namely:

- Present a brief background of the problem
- Discuss the problem targeted by Web framework.
- Ascertain the root causes of the problem.
- Define the methodology.
- In addition, recognize the constraints imposed on the solution.

Chapter II details the selection of an XML based Web framework, which provides the architectural components of the application server. The chapter covers the challenges of building a flexible, extensible, and scaleable system that allows efficient data integration and semantic interoperability.

Chapter III contains a review of techniques and technologies proposed to implement the Web framework. Along with the advantages and disadvantages of various state-of-the-art technologies are detailed and commercially available tools.

Chapter IV applies XML and Java related technologies to connect to heterogeneous database for analysis.

Chapter V automates the development of XSL stylesheets to enable translations between XML and 3D system object representations

Chapter VI provides concluding remarks about the current system state. In addition, this chapter will review potential work for further research.

THIS PAGE INTENTIONALLY LEFT BLANK

II. WEB PUBLISHING FRAMEWORKS

A. INTRODUCTION

This chapter describes the most significant applications of XML. While there are literally hundreds, and soon to be, thousands of important applications of XML, the topics in this chapter have a significant potential to change the way traditional Web site development processes occur. Using XML for presentation styling is very important, however XML data portability is the major justification for an XML-based application server. XML provides the format for the exchange of information between heterogeneous systems and a means of providing interoperability within DoD. The Web-publishing framework separates the presentation from content and thus facilitates the exchange of data between heterogeneous systems. This thesis supports a framework, which allows for the connection and conversion of a non-XML legacy system to an XML-based system. The framework also allows for the conversion of XML data into a wide spectrum of different presentations.

B. OVERVIEW

The US military is moving closer to a completely Web-based front-end for imbedded systems used to support interoperability. The United States Message Text Format (USMTF) has long been the tactical messaging tool of choice for the Department of Defense. As the Extensible Markup Language (XML) continues to proliferate and gain momentum in the commercial sector, many of its proponents have questioned the usefulness of the USMTF system. An Extensible Markup Language Message Text Format (XML-MTF) Development Team is converting the USMTF format to an XML-MTF format. While work is ongoing, tools are now available to take a common USMTF message and convert it into XML-MTF messages.

At the same time, the military needs more functionality, and interoperability between forces. The result has been the rise of the Web framework, which can support common data exchange and a separation between presentation and software coding using the MVC approach. As this framework is deployed, the new role of presentation is different from the Webmaster, in that little or no Common Gateway Interface (CGI) script, Active server Pages (ASP), JavaScript, or other scripting language code are part of the job description. Instead, the Web presentation may focus solely on the Web site's presentation, comprised of HTML and Format Object (FO) creation, modification, and development. The rapid changes in the military environment may require a complete application or site overhaul, which, in turn, may include changing hundreds of HTML pages to support joint operations. While Cascading Style Sheets (CSS) have improved, the difficulty of maintaining consistency across these pages requires a huge amount of time. Currently, even if this less than ideal situation were acceptable, software developers in DoD are a sparse commodity, and spending long periods of time making markup language changes to Web pages is an inefficient allocation of resources. This process may also lead to large time delays in page updates or Web site version changes. The bottom line is that Web sites need to be more flexible than they currently are.

With the advent of server-side Java, the problem has only grown. Servlet developers find themselves spending long hours modifying system.out.println() statements to output HTML, which in turn may require the software developer to modify Java code (Figure II.). The entire Java Server Pages (JSP) specification arguably stemmed from this situation; however, JSP is not a solution, as it only shifts the frustration to the HTML developer, who constantly has to avoid making incidental changes to embedded Java code. In addition, JSP does not provide the clean separation between content and presentation that it promises. What is required is a means to generate pure data content, and have that content uniformly styled either at predetermined times (static content generation), or dynamically at runtime (dynamic content generation).

The XSL and XSL Transformations (XSLT) technology space will improve the transformation of data. The problem, which remains, is that an engine must exist to handle content generation, particularly in the dynamic sense. Having hundreds of XML documents on a site has little benefit if there is no mechanism to apply transformations to them when requested. An additional requirement is for Servlets and other server-side components to output XML that should be consistently styled. This defines the requirements for the Web-publishing framework, which will be described in this chapter.

This Web-publishing framework attempts to address the complicated issues raised above. Just as a Web server is responsible for responding to a Uniform Resource Locator (URL) request for a file, a Web-publishing framework is responsible for responding to a similar request; however, instead of responding with a file, it often will respond with a published version of an XML file. A XML file is transformed using XSLT stylesheet; the document is processed within the application level, or converted into another format such as a VRML or Portable Document Format (PDF) output file. The requestor does not see the raw data that may underlie the published results, but also does not have to explicitly request which publication occurs. Often, a Uniform Resource Indicator (URI) base (such as http:/www.xmlking.com/cocoon) signifies that a publishing engine that sits on top of the Web server should handle requests. The concepts are much simpler to describe than to implement within a framework, and finding the correct framework for Web site development is a nontrivial task.

C. SELECTING A FRAMEWORK

The available publishing frameworks, which support the requirements, are limited (see Appendix A). One of the best resources currently available products is XML Software's list at http://xmlsoftware.com/publishing/. An evaluation of stable frameworks (see Appendix A reduces the list substantially). [XmlSof01] Appendix A provides a break down of systems to consider in the selection of the Web publishing frameworks.

1. Product Stability

Most of the sites listed in Appendix A are still in the early development phase. In fact, only fifteen systems are available which support a second-generation framework. While a higher version number is not a guarantee of stability, it often reflects the amount of time, effort, and review that a framework has undergone. The XML publishing systems are a new beast resulting in the market being flooded with version 1.0 and 1.1 products that simply are not stable enough for research at this time.

Our criteria for selecting framework tools are provided below. If the framework tools do not offer Simple API for XML (SAX) 2.0 and Document Object Model (DOM) Level 2 support, or the product is a version 1.0 and 1.1 products or beta product, the product was not considered as a possible framework for this research. Also not considered were platform-specific technology and proprietary systems. These frameworks do not provide heterogeneous solution for easy of use and portability across multiple platform. Frameworks under consideration supported heterogeneous deployment across multiple platforms and did not lock DoD into a specific platform and/or operating system (such as Windows only). The goal of this thesis was to find a framework, which served clients on any platform and not to add a new stovepipe to the current Navy arsenal of legacy problems.

2. XML Tools and API Support

Stable frameworks, which were identified, supported a variety of XML parsers and processors. Frameworks using a platform specific parser or processor are the XML equivalent of a Windows only system. Although frameworks often integrate well with a particular parser vendor determining, a parser's interchangeability is important to support multiple types of XML documents[Hun00].

Support for SAX and DOM is an important indication of the framework's longevity. Other frameworks for consideration were those whose developers were monitoring and incorporating the specifications of XML Schema, Xlink, XPointer, and other emerging XML technologies. The framework revisions often indicated added support for these XML specifications.

3. Heterogeneous Environments

The Web framework uses XML as the interoperability language for data integration. The use of Java's Web technologies is not only the best language for integration with XML, but also the foundation for most of the open-source XML, Servlets, editors, parsers, validators, and tools. Java tackles many of the heterogeneous environment problems. Java supports applications in a heterogeneous environment, which requires that applications be capable of executing on a variety of hardware architectures and operating systems. This is achieved by generating an intermediate code called byte code, which has an architecturally neutral format designed to be transported easily to multiple hardware and software platforms. Such a design can relieve both the developer and user of concerns related to heterogeneity of the target platforms. In addition, another attractive feature of Java is the notion of downloaded executable content (data that contain programs that are executed upon receipt). Such designs will avoid the software distribution problem and add new capabilities to the system implemented in a new Java class. [DIAWS01] Java provides unlimited possibilities to enhance the function of the Web-publishing framework of the future.

4. Market Presence

The last, and perhaps most important, question answered was determining if a Web publishing framework is used in production applications. Frameworks which lacked vendor reference applications or site locations were not considered. Vendors and developers should be able to support implementation of the products and proud to display their frameworks in action.[Hun00] We interpreted hesitance in this area as a sign that the framework was not ready for the production environment.

5. Making the Decision

After examining several selections, only a few sites, exhibited stable, heterogeneous, integration with multiple XML Tools and API, and provided support in a production environment. As of the date of this research (November 2001), less then ten publishing frameworks existed that support the latest version of SAX (version 2.0), DOM (Level 2), and Java API for XML Processing (JAXP) (Version 1.1), which are in production and have at least three significant revisions of code under their belts in development. This will undoubtedly change over time as Web publishing frameworks evolve.

Cocoon	
Version: 2.0 release candidate 1	Good
Developer: Apache Software Foundation	
License: Open source	Good
Platforms: Java	Good
DOM 2.0	Good
Multiple Parser	Good
Supports July XSL draft	Good
SAX 2.0	Good
JAXP 1.1	Good
Publishing framework HTTP Servlet that supports the use of DOM, XML, and XSL to provide Web content. The content is created in XML conform a DTD. Cocoon can then process the XML and apply an XSL stylesheet to it for rendering (different stylesheets for different browsers can be used. You can use Cocoon, for example, to create well-formed HTML pages from an XML source. Includes a SAX API and XSLT stylesheet compilation.	

Table II.1 Cocoon Web Publishing Framework

Of these intial candidates however, Apache Cocoon Web Publishing Framework (Table II.1) was the only one which has demonstrated consistent success within the Java and XML community. Cocoon is an XML Application Server-based program which allows any conformant parser to be used, and is based on the Java Servlet technology. The system also is part of the Apache XML project, and has default support for Apache Xerces and Apache XALAN. It allows any conformant XML parser to be used, and is based on the Java Servlet architecture. In addition, there are several production sites using Apache Cocoon (in its 1.8 form). For these reasons, we have selected Cocoon as our underlying framework[Hun00].

D. OBJECTS

Cocoon is a publishing framework designed using Java. Cocoon publishing framework uses W3C technologies such as DOM, XML, and XSL to provide Web content. The Cocoon supports document content, style, and logic created by different system, individuals, or working groups. Cocoon's layered framework provides for independent designs, reducing time and management overhead, and increasing work reuse.

Cocoon separates different XML files, and uses XSL transformation capabilities to merge them into a presentation or a new XML output. Cocoon provides the user with different ways of working, allowing independent content, logic, and style processes [Axm01].

E. INTEROPERABILITY

Cocoon Web publishing framework will support the Navy's XML infrastructure and provide "Web Enablement" application services. The implementation of interoperable Web technologies across the Naval infrastructure allows subscribers and publishers of information to pull or push information "services" as required to performed operational or business transactions. [WEN01] Open source code is available for Cocoon Web Publishing Framework. The Web-publishing framework can provide information and Web transaction to multiple databases either XML based or not. Chapter IV will demonstrate some of the connection methods and technologies. Further, this research provides a mechanism for interoperability with joint/coalition forces and a means of conducting business with external agencies.

This research demonstrates the design and implementation of a Web-publishing framework, which exploits the advantages of Java to construct the framework, and which provides the solution for the automation of many tasks. The system also provides excellent high availability features, including automatic detection of failures, and alerts in the form of log files. In addition, the Java approach enables the building of a high performance server using a variety of hardware and software components.

F. MODEL

This Web publishing framework provides a means for resolving these differences in interoperability and representation with a simple, yet powerful, model as seen in figure II.1. The Cocoon model allows Web sites to be highly structured and well designed, reducing duplication efforts and site management costs by allowing different presentations of the same data depending on the requesting client (HTML clients, PDF clients, Wireless Markup Language (WML) clients, X3D Simulation clients), and by separating according to different contexts requirements, skills and capacities. Cocoon enables better human resource management by reducing to a minimum the cross talks between different working contexts.

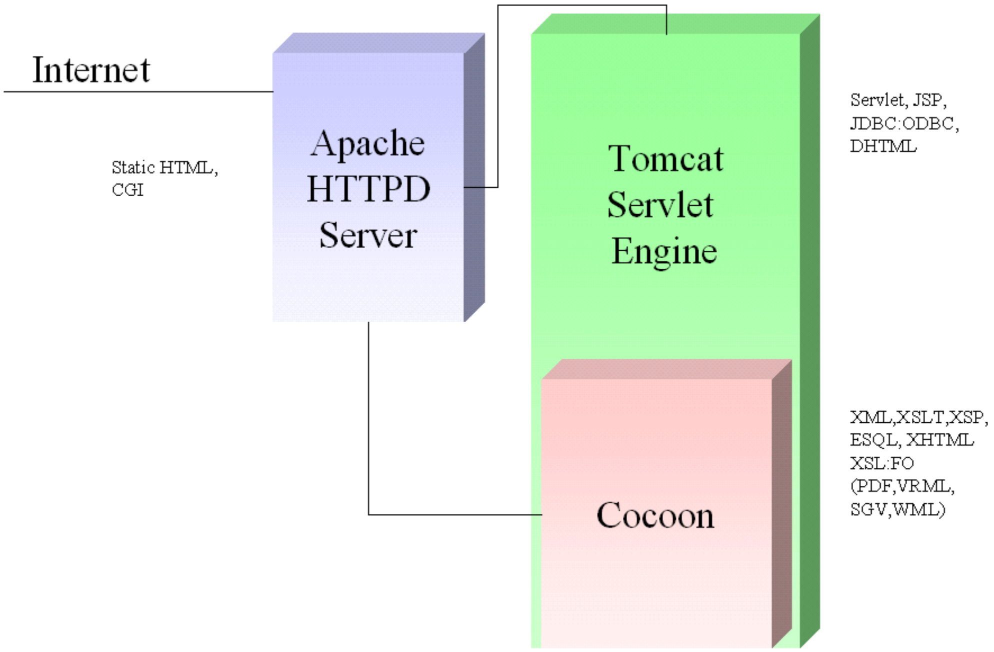


Figure II.1 Cocoon Web Publishing Framework

To do this, the Framework model divides the development of Web content into three separate levels:

- **XML creation** - the content owners create the XML file. They do not require specific knowledge on how the XML content is further processed rather than the particular chosen DTD/namespace.
- **XML processing** - the requested XML file is processed and the logic contained in its logicsheet is applied. XML separates logic from the content file unlike other dynamic content generators.
- **XSL rendering** - the created document is then rendered by applying an XSL stylesheet to it and formatting it to the specified resource type (HTML, PDF, VRML, XML, WML, XHTML)

The Web-publishing framework concentrates on solving the XML publishing infrastructure problem. In that respect, it is ahead of many of the vendors who worry primarily about the technology needed and less about how it integrates into an XML-based infrastructure. [Axm01]

G. CONSTRUCTION OF A WEB PUBLISHING FRAMEWORK

Enabling the sharing of information has the potential for significantly enhancing the capability of the XML based Web-publishing framework compared with individual components. The introduced framework enables information sharing among autonomously developed heterogeneous systems. Appendix B is an Installation guide for the Cocoon Web Publishing Framework on Window 2000. Construction of this Web framework affords the following benefits:

- The ability to discover the information and operations shared between server components.
- Assistance in identifying the different representations used for such information and operations by component systems.
- Definition of the transformations required to translate between different representations.
- CGI Scripting using Apache and generation of system-specific information used to resolve representational differences between component systems.
- Heterogeneous connections between multiple RDBMS Java Database Connection (JDBC): Object Database Connection (ODBC) bridges.
- Identification of the capabilities which support the WEN using the Navy XML Infrastructure as in the Web-publishing framework.

- Automation of the development of XSL stylesheets to enable translations between X3D and VRML system object representations

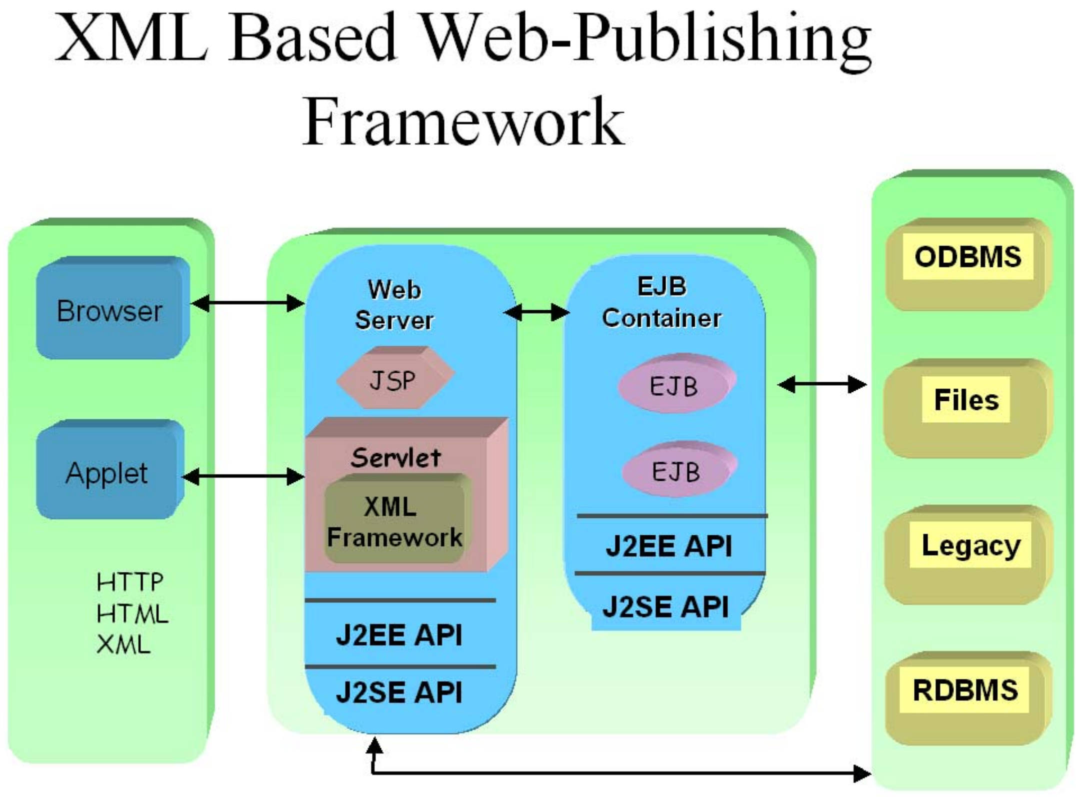


Figure II.2 Cocoon Web Publishing Framework

1. Apache Web Server

The first task in constructing Web framework in Figure II.2 is to setup and load the Apache Web Server. Apache is a Hyper Text Transfer Protocol (HTTP) server, originally designed for Unix systems. Apache recently extended its Web server to support Microsoft Windows 2000 operating system[Ahttp01].

Apache uses the C++ program language to program the Apache Web server. Apache is also open source Web server and provides both source code and executable files for listed operating system. Apache provides the following desirable features:

- **HTTP/1.1 Compliance**

Aside from the optional proxy module (which operates as HTTP/1.0), Apache is conditionally compliant with the HTTP/1.1 proposed standard, as approved by the Internet Engineering Steering Group (IESG) and the Internet Engineering Task Force (IETF) HTTP working group. HTTP/1.1 provides an improved protocol, and allows greater performance and efficiency when transferring files. Apache still works well with HTTP/1.0 browsers.

- **Extended Server Side Includes (XSSI)**

A new set of server-side include directives allows the user to create better World Wide Web (WWW) pages. This includes a number of powerful new features, such as the ability to set variables and use conditional HTML.

- **File-Based And REGEX-ENABLED Directive Sections**

The new <Files> section allows directives to be enabled based on full filename, not just directory and URL. In addition, <Files> sections can appear in htaccess files. <Files>, along with <Directory> and <Location>, can now be based on regular expressions, not just simple prefix matching.

- **Browser-Based Environment Variables**

Environment variables can now be set based on the User-Agent string of the browser. Combined with XSSI, this allows the user to write browser-based conditional HTML documents.

- **SETUID CGI Execution**

Apache supports the execution of CGI scripts as users other than the server user. Apache has built in security checks to provide a safe environment.

- **URL Rewriting Module**

The optional mod_rewrite module is included. The module can provide powerful URL mapping, using regular expressions. Apache provides many advanced features.

- **Enhanced, Configurable Logging**

The optional mod_log_config is included with Apache and is enhanced to allow logging of detail about the transaction, and can be used to open more than one log file at once (each of which can have a different log format). Apache can write any logs to a directory, which is writable by anyone other than the user that starts the server.

- **User Tracking (Cookies) Revisions**

Apache supports the use of cookies directly. The mod_usertrack provides the user with the ability to disable the generation of cookies and can disable pre-compiled cookie modules. In addition, an expiration time can be set on the cookies.

- **Virtual Host**

The Virtual Host directive works with more than one IP address or hostname. This lets a single vhost handle requests for multiple IPs or hostnames. In addition, the special section <VirtualHost _default_> provides request handling.

- **CGI Debugging Environment**

Script Log allows the user to set up a log that records all input and output to failed CGI scripts. This includes environment variables, input headers, POST data, output, and more. This makes CGI scripts much easier to debug.

- **Resource Limits For CGI Scripts**

New directives allow the limiting of resources used by CGI scripts (e.g., max Central Processing Unit (CPU) time). This is helpful in preventing 'runaway' CGI processes.

- **Redirect Directive Can Return Alternate Status**

The Redirect directive can return permanent or temporary redirects, "Gone" or "See Other" HTTP status. Apache also implements National Center for Supercomputing Applications (NCSA) -compatibility, RedirectTemp and RedirectPermanent.

- **Add Or Remove Options**

The Options directive can now add or remove options from those currently in force, rather than always replacing them.

- **Command-Line Help**

The -h command-line option now lists all the available directives.

- **Optional Headers Module To Set Or Remove HTTP Headers**

The optional mod_headers module can be used to set custom headers in the HTTP response. Apache can append to existing headers, replace them, or remove headers from the response.

- **NCSA Satisfy Authentication**

The National Center for Supercomputing Applications, one of the five original centers in the National Science Foundation's Supercomputer Centers Program and a unit of the University of Illinois at Urbana-Champaign. NCSA greatly broadened the user base of remote supercomputing and the Internet with NCSA Telnet in 1987. In 1992, the center introduced NCSA Mosaic, the first readily-available graphical Web browser.[NCSA] Apache server supports directive implemented satisfy NCSA authentication and allows for more flexible access control configurations.

- **NCSA Compatibility**

Apache directives are more compatible with NCSA 1.5 to make moving between servers easier. In particular, Apache implements the Satisfy, MaxKeepAliveRequests, Redirect Permanent and RedirectTemp, directives, and the following directives are now syntax-compatible with NCSA: AuthUserFile, AuthGroupFile, AuthDigestFile, KeepAlive and KeepAliveTimeout.

- **Optional proxy module**

An improved File Transfer Protocol (FTP), HTTP, and CONNECT mode Secure Socket Layer (SSL) proxy is included.

2. Tomcat Servlet Engine

The next step in constructing the Web framework is to install the Jakarta-Tomcat Servlet Engine after the Apache Web Server. Tomcat Servlets on the server side are analogous to applets on the client side. The Web server executes Servlets inside the Java Virtual Machine (JVM). Figure II.3 shows the Servlet engine and Web server relationship. Major Web servers, including the Netscape Web sever, Microsoft’s Internet Information Server (IIS), the W3C Jigsaw Web server, and Apache Web Server incorporate Servlets technology[Tom01].

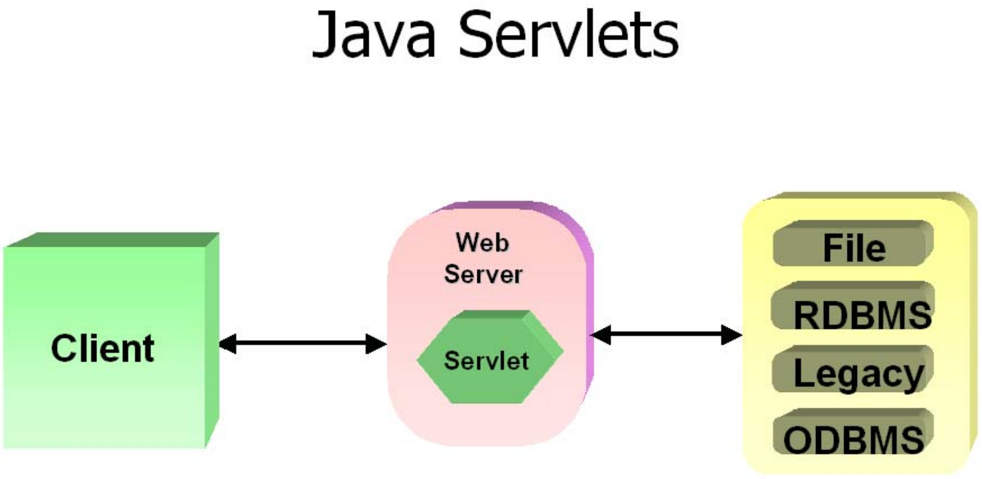


Figure II.3 Java Servlet Engine and Web Server Relationship

Servlets are Java Technology's answer to CGI programming. Servlets are Java programs that run on a Web server, acting as a middle layer between a request coming from a Web browser or other HTTP client and database or applications on the HTTP server. Java programs offer a proven structured programming and object-oriented programming (OOP) environment [Dietel]. The Tomcat Servlet Engine uses the Java programming language to program the Servlet engine. Tomcat is also an open source Servlet engine and provides both source code and binary files which can be used in any operating system containing a JVM. The Tomcat Servlet Engine using Servlets executes the following steps [Tom01].

- Read data sent by the user. Servlets are invoked and executed on the server. The data is usually entered in a form on a Web page, Java applet, Java Servlet, or a custom HTTP client program.

- Servlets receive a request from the client and Look up any other information embedded in the HTTP request. This information includes details about browser capabilities, cookies, the host name of the requesting client, and so forth.
- Generate the results. This process may require talking to a database, executing a Remote Method Invocation (RMI) or COBRA call, invoking a legacy application, or computing the response directly.
- Format the results inside a document.
- Set the appropriate HTTP response parameters, telling the browser what type of document is being returned, setting cookies, caching parameter and other such tasks.
- Servlets are very powerful and written by software developers. Servlets provide for structured programming using an object-oriented language.

3. Cocoon Publishing Framework

Cocoon is a Java publishing framework that relies on new W3C technologies (such as XML and XSL) to provide Web content. Cocoon is also open source publishing framework and provides both source code and binary files capable of operating with any operating system, which has a JVM. The Cocoon processing model in figure II.4 provides the separation of:

- Production - where XML content is generated based on Request parameters (Servlet equivalent)
- Processing - where the produced XML content is transformed/evaluated
- Formatting - where the XML content is finally formatted into the desired output format for client use.

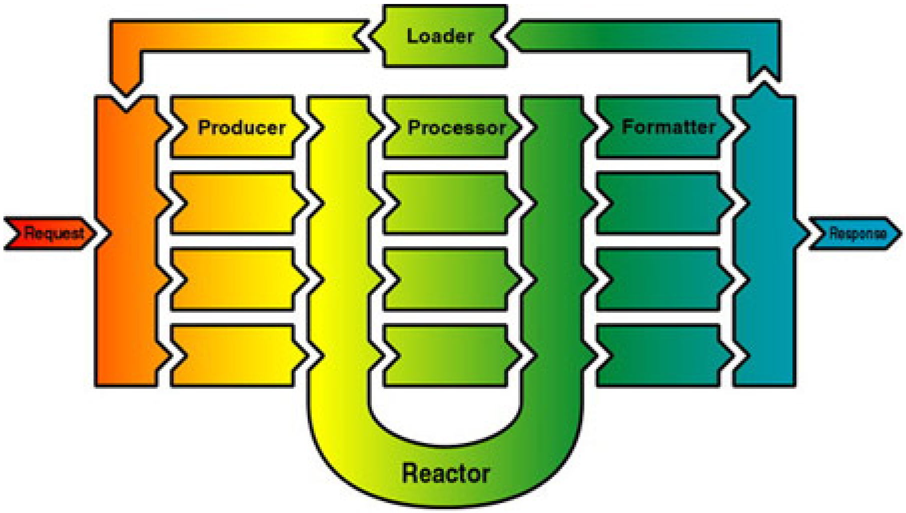


Figure II.4 Cocoon processing model

This separation of working contexts allows Cocoon users to implement their own internal modules to add the functionality require to tailor a publishing system. Cocoon distribution includes a number of processors that implement common needs and situations. These are:

- **XSLT processor** - Applies XSLT transformations to the input document. XSLT allows you to solve your transformation needs as well as simple tag evaluation/processing due to its extensible and programmable nature. XSLT is a W3C Recommendation.

- **Extensible Server Pages (XSP) processor** - Evaluates XSP pages and compiles them into producers. This processor allows you to include programmatic logic into your pages as well as to separate the logic from the content. Note that the XSP Processor assumes that it is getting its input from a static file, so it will not work well with pre-processing. Its design indicates that it should really have been a Producer in the first place, instead of a Processor. XSP also evaluates XML processing instructions with multi-language (Java and EcmaScript) logic. This processor allows you to do programmatic substitution and inclusion eliminating the need for complex processing logic.

- **Extensible Standard Query Language (ESQL) processor** - Evaluates simple tags describing ESQL queries to JDBC drivers and formats their result-set in XML depending on given parameters. Extended SQL taglib allows easy post-processing of output within XSP.

- **ESQL logicsheet** is an XSP logicsheet that performs SQL queries and serializes their results as XML. This allows the user to work with data from a wide variety of different sources when using Cocoon.

- **Lightweight Directory Access Protocol processor** - Lightweight Directory Access Protocol (LDAP) provides access to the X.500 Directory while not incurring the resource requirements of the Directory Access Protocol (DAP). This protocol is specifically targeted at simple management applications and browser applications that provide simple read/write interactive access to the X.500 Directory, and is intended to be a complement to the DAP itself. Key aspects of LDAP are protocol elements carried directly over TCP or other transport bypassing much of the session/presentation overhead. The *LDAP* processor evaluates simple tags describing LDAP queries to directory services, and formats their result-sets in XML depending upon given parameters. LDAP Processor is a processor for Cocoon that performs LDAP queries, translates the result set into an XML fragment, and inserts the fragment in the original document.

This research has presented the design and implementation of an XML based Web-publishing framework. In the next chapter, we will describe XML and JAVA technologies to leverage the advantages of an XML-based system.

THIS PAGE INTENTIONALLY LEFT BLANK

III. TECHNOLOGICAL INTEROPERABILITY ENABLERS

A. INTRODUCTION

This chapter discusses the technologies implemented in the Cocoon Web Publishing Framework. Each discussion focuses on a description of the technology, implementations of that technology, and why the system uses that particular technology. There are three major factors that drove the design decisions for the implementing the XML-based publishing framework: DoD constraints,technology requirements for interoperability, and cost reduction.

1. Directives

The DoD is counting on XML and XML-related technologies to enable information dissemination management capabilities and to solve many interoperability issues. DoD Directive 8320.1, DoD Data Administration [DoD91], authorizes the establishment of, and assigns responsibilities for, DoD data administration to plan, manage, and regulate data within the Department of Defense. The Defense Information Systems Agency (DISA) is the lead agency responsible for executing the policy and procedures and making DoD Data Standards available to the community. DISA is using XML as its common exchange data format in support of its Defense Information Infrastructure Common Operating Environment (DII COE) data engineering strategy. The bottom line is that Navy XML Infrastructure (NXI) provides information exchange between the United States Navy and DoD systems.

2. Web Frameworks Technology Needs

The Web Frameworks need to support many different types of technologies, includng XHTML, VRML, DHTML, JSP, XSP, JAVA, SERVLET, and XML. XML and its related technologies are not sufficient as stand alone systems. Java with its strong typing, properties, methods, interface inheritance, behavior inheritance, polymorphism, reflection, and its compatibility with XML, made its use essential. In addition, XML technologies like DTD and XSLT are compatible within an XML based Web Publishing Framework.

3. Cost Reduction

Reducing the cost of doing business is one of the major benefits of this project. In this vein, the Cocoon Web Publishing Framework reduces the cost of actually building the system itself. The biggest reduction results from the liberal reuse of open-source software. Open-source software reduces costs in four ways:

1. It is free!
2. The world can read, modify, and even redistribute this type of software. This allows users to leverage and build on the work of others. This reduces the cost of labor in the software development.
3. Because of the second point the system is assured that the software incorporated will continue to evolve while at the same time exposing the code to testing by the masses. Again, this is a savings in labor with regard to testing and maintenance.
4. The life cycle for this type of software is extremely compressed, i.e., better software faster, when compared with proprietary software. This is possible because many companies and individuals are all able to contribute to the development of software.

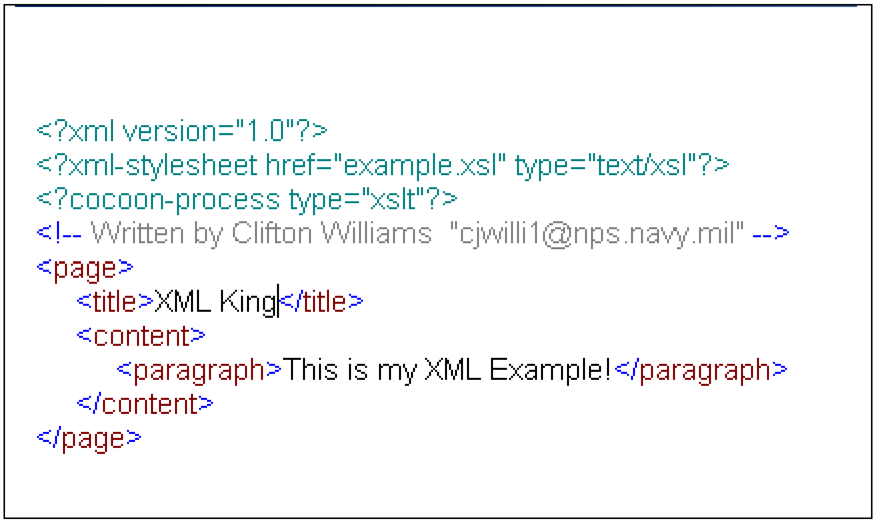
B. XML THE NEW LINGUA-FRANCA

XML stands for Extensible Markup Language, which is a meta-markup language that became a World Wide Web Consortium (W3C) recommendation in January 1999. It is a new language for structured documents and data on the Internet. XML is platform-independent, non-proprietary, customizable, self-describing and human readable. This chapter assumes the reader has passing familiarity with this topic and will not provide an in-depth discussion on XML[Chri01]. See [W3C] for more detailed information.

XML is a family of technologies. This research will focus on a few members of this family, namely XML, DTD, XML Schema, SAX, DOM, XSLT and XHTML.

1. XML basics

The best way to get a feel for XML is by viewing an example. Figure III.1 below shows a simple XML document.



```
<?xml version="1.0"?>
<?xml-stylesheet href="example.xsl" type="text/xsl"?>
<?cocoon-process type="xslt"?>
<!-- Written by Clifton Williams  "cjwilli1@nps.navy.mil" -->
<page>
    <title>XML King</title>
    <content>
        <paragraph>This is my XML Example!</paragraph>
    </content>
</page>
```

Figure III.1 Simple XML Document

XML looks similar to HTML. The major difference is that XML is a meta-language that offers no inherent clue as to how the information should look. This frees XML from the static tag set of HTML and separates the model created from its view. All XML documents are properly nested (hierarchical) tree structures. This example document contains a root element </>, which contains one child element <pages>, which, in turn, contains two child elements called <title> and <content>. Notice that the XML document nicely describes the structure of the data.

2. Well-formedness

A document is not an XML document unless it is well formed, i.e., syntactically corrects, according to the W3C's XML specification. Ill-formed documents are not accepted for processing. This simplifies the internal code of parsers and speeds up the processing of documents.

3. Validity

An XML document is valid if it has an associated schema, DTD or XML Schema, and if the document complies with that schema. A schema further constrains the syntax of the XML document and adds semantics through documentation within the schema itself.

C. CONSTRAINING CONTENT

1. DTD

A Document Type Definition (DTD) specifies the logical structure of the document; it is a formal grammar describing document syntax and semantics. The purpose of a DTD is to define the legal building blocks of an XML document. It defines the document structure with a list of legal elements. A DTD can be declared inline in your XML document, or as an external reference. With DTD, each of your XML files can carry a description of its own format with it. With a DTD, independent groups of people can agree to use a common DTD for interchanging data. Applications can use a standard DTD to verify that the data receive from the outside world is valid. The user can also use a DTD to verify is own data. This thesis does not discuss DTD in significant detail because the logical structure and designs of DTD used in the Web framework were already defined an incorporated. The DTD's biggest flaw is that it has no capability for strong typing. DTDs treat almost all of its data as simply strings[Chri01].

2. XML Schema

An XML Schema formally specifies a coherent set of definitions for elements and attributes. Within the schema definition are declarations of which elements, attributes etc. are legal within a document, and the locations where they are legal. [Gold01] In other words, the schema specifies the content and structure of an XML document. Figure III.2 show a simple yet powerful example of a XML schema. If a document does not conform to the XML schema specification, it is not valid.

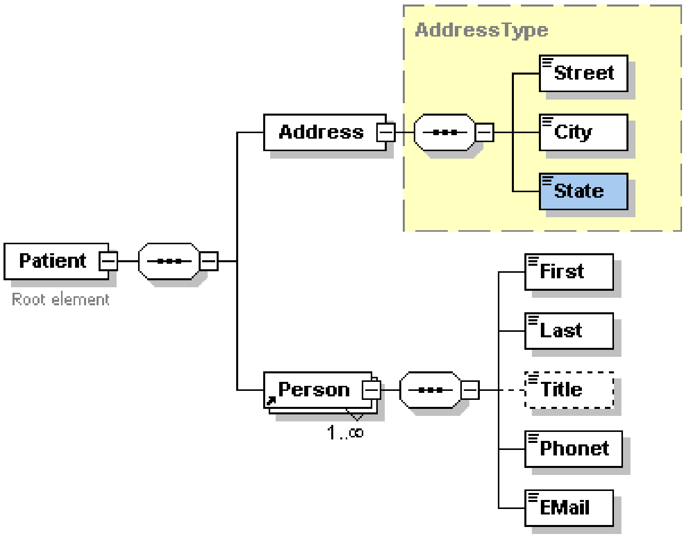
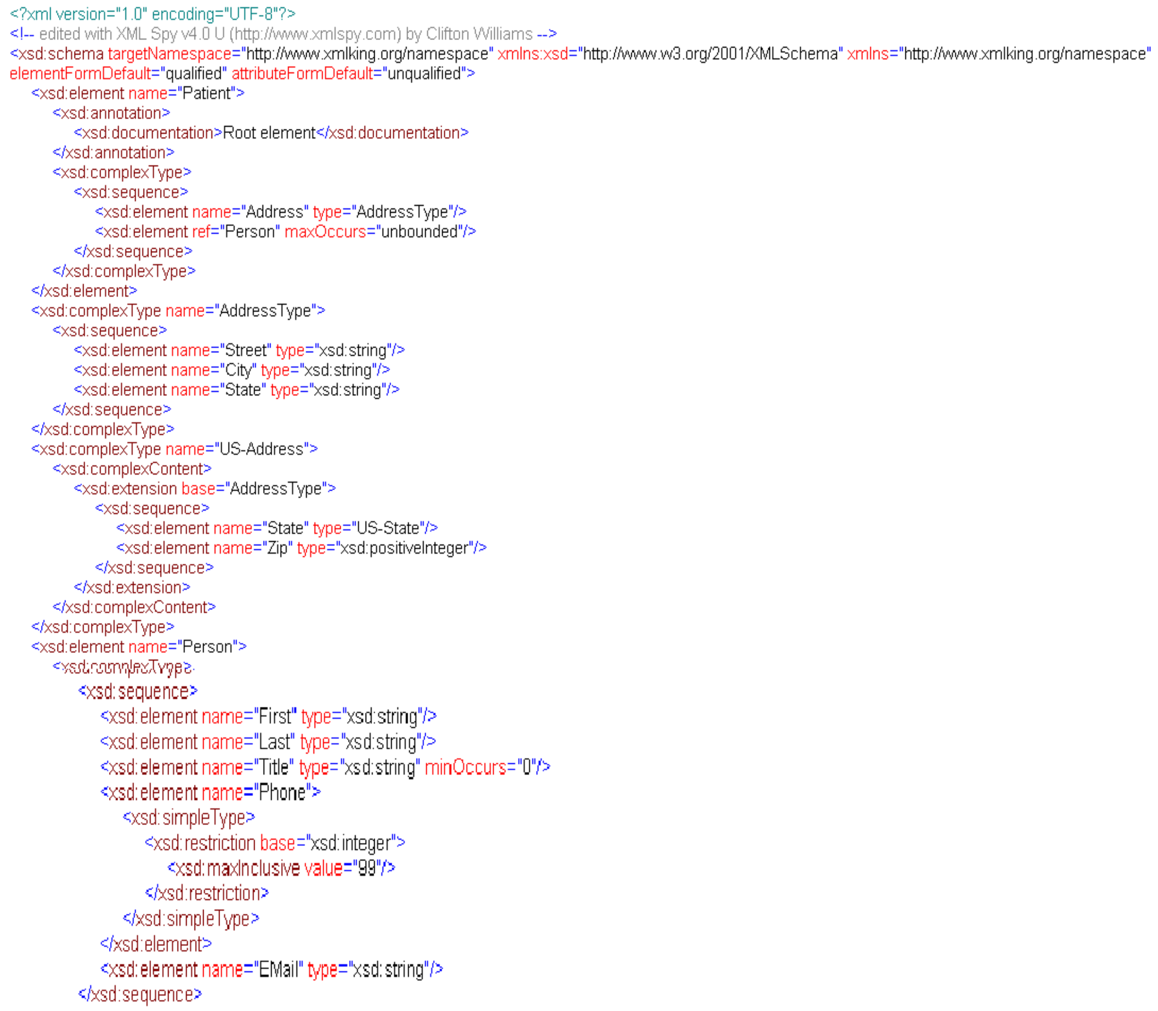


Figure III.2 XML Schema as designed in XMLSPY

The XML Schema is a W3C recommendation that improves the deficiencies of the Document Type Definition (DTD). DTDs also specify the content and structure of an XML document, but DTD document schemas are not written in XML and only support character data. XML schemas make up for these deficiencies while providing new capabilities that make it more robust and powerful than a DTD. Some of the extensive additional capabilities that schemas as seen in figure III.3 provide include:

- New built-in data types such as integers and dates;
- User-defined data types that can be created to put additional constraints on built-in data types;
- Any non-negative whole number can be used to specify the number of child element occurrences (formerly just zero, one, and infinity were allowed);
- Schema constructs can be imported from existing schemas;
- A process called *refinement* enables elements to inherit content and attributes from other elements. [Bird99]



```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.0 U (http://www.xmlspy.com) by Clifton Williams -->
<xsd:schema targetNamespace="http://www.xmlking.org/namespace" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://www.xmlking.org/namespace"
elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xsd:element name="Patient">
        <xsd:annotation>
            <xsd:documentation>Root element</xsd:documentation>
        </xsd:annotation>
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="Address" type="AddressType"/>
                <xsd:element ref="Person" maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:complexType name="AddressType">
        <xsd:sequence>
            <xsd:element name="Street" type="xsd:string"/>
            <xsd:element name="City" type="xsd:string"/>
            <xsd:element name="State" type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="US-Address">
        <xsd:complexContent>
            <xsd:extension base="AddressType">
                <xsd:sequence>
                    <xsd:element name="State" type="US-State"/>
                    <xsd:element name="Zip" type="xsd:positiveInteger"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:element name="Person">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="First" type="xsd:string"/>
                <xsd:element name="Last" type="xsd:string"/>
                <xsd:element name="Title" type="xsd:string" minOccurs="0"/>
                <xsd:element name="Phone">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:integer">
                            <xsd:maxInclusive value="99"/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:element>
                <xsd:element name="EMail" type="xsd:string"/>
            </xsd:sequence>
```

Figure III.3 XML Schema

An XML Schema can specify actual data types for each element's content within a document. In addition, XML Schemas have built in elements to handle annotations to add semantics to the types created. XML Schemas can handle multiple namespaces allowing a means to process multiple XML documents. A more detail discussion of XML Schema can be found at [W3C].

D. PROGRAMMATIC ACCESS

1. SAX

Simple API for XML (SAX) and DOM (Document Object Model) were both created to serve the same purpose of providing access and modification capability to the information stored in XML documents via any programming language. However, both of these techniques take very different approaches in providing that access.

SAX provides access to documents as a sequence of events. It works as follows:

- A SAX parser sequentially processes an XML document, signaling an event when a specified item such as an open tag or close tag is found.
- The programmer is responsible for interpreting these events by writing an XML document handler class. This handler class is responsible for specifying what actions are required when a tag is encountered, such as storing an element for future reference.

2. DOM

XML only supports "has a" or "parent-child" relationships, such as a <patient> may contain sub-elements of <first_name>, <middle_name>, <last_name>, <title>, <social_security_number>, etc in a hierarchical tree structure. The Document Object Model (DOM) preserves the hierarchical tree structure. DOM creates a tree of nodes based on the structure and information in an XML document, and provides access to this information by interacting with this tree of nodes. The DOM takes a generic approach, in that it will take any arbitrary XML document and model it. The DOM creates a document object tree (by the XML parser, or by internal coding). Using the interfaces for the DOM API, a developer can access elements in the tree, modifying, deleting, and creating leaves and branches.

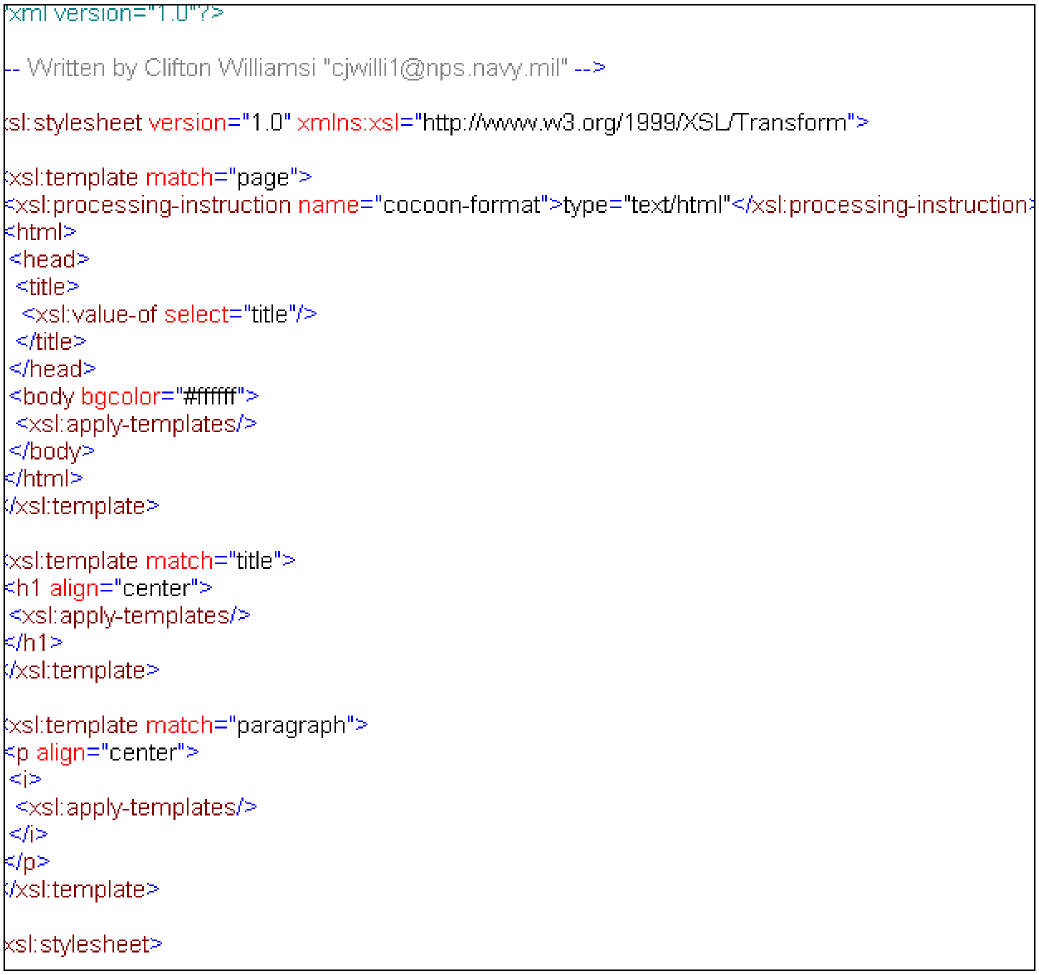
3. SAX vs. DOM

The choice of SAX or DOM is depends upon how complex a document the programmer wishes to access, ease of use, and performance concerns. The SAX treats a document as a series of events, which means it efficiently and swiftly analyzes large XML documents. The drawback is that the developer has to define the data structure to the hold element data. The DOM must load the entire document in-memory before granting access. This takes more memory and time compared to SAX. The DOM's primary advantage is that the parser does almost everything, including reading the XML document in, creating a Java object model on top of it, and then giving a reference to this object model (a Document object) for manipulation. Within the Web framework, speed and memory are critical issues so the SAX will usually perform better, however DOM's ease of use makes it preferable when development time is a major issue[Chri01].

E. TRANSLATIONS EXTENSIBLE STYLES LANGUAGE (XSL)

The Extensible Stylesheet Language (XSL) is a W3C specification for applying standard formatting to XML documents. It consists of two standards: format and transformation. The formatting standard defines the semantics of formatting in the form of a catalog of formatting objects. The transformation standard has evolved into a language that is autonomously useful for transforming one XML document to another. It is known as Extensible Stylesheet Language for Transformations (XSLT). XSL is the link between abstract data required for computer processing and the formatted rendition essential for human readability. XSLT transforms a document from an abstraction to a rendition. (Goldfarb, 2001) By using multiple XSL stylesheets, a single message can be formatted for text, XHTML, printing, or e-mail.

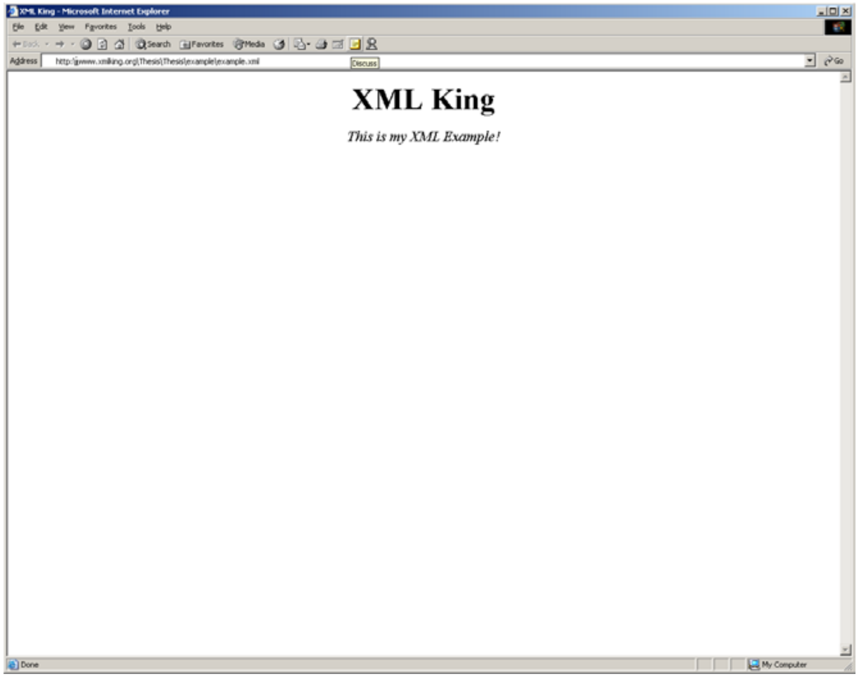
Extensible Styles Language (XSL) is essentially three languages: a transformation language (XSLT), an accessing language (XPath), and Extensible Markup Language Formatting Object (XML-FO).



```
xml version="1.0"?>

-- Written by Clifton Williamsi "cjwilli1@nps.navy.mil" -->

sl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

xsl:template match="page">
<xsl:processing-instruction name="cocoon-format">type="text/html"</xsl:processing-instruction
<html>
 <head>
  <title>
   <xsl:value-of select="title"/>
  </title>
 </head>
 <body bgcolor="#ffffff">
  <xsl:apply-templates/>
 </body>
</html>
/xsl:template>

xsl:template match="title">
<h1 align="center">
 <xsl:apply-templates/>
</h1>
/xsl:template>

xsl:template match="paragraph">
<p align="center">
 <i>
  <xsl:apply-templates/>
 </i>
</p>
/xsl:template>

xsl:stylesheet>
```

Figure III.4 Simple Example Stylesheet

XSLT is a high-level, declarative, and XML-based language. It allows a programmer to write XSLT code (style sheets) that transforms an XML document into any text-based document. Figure III.4 is a simple Example XSLT style sheet and Figure III.5 is the associated browser display.


XML King - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www.xmlking.org\Thesis\Thesis\example\example.xml
Discuss
Go
Done
My Computer

XML King

This is my XML Example!

Figure III.5 Simple Example Browser Display

XSLT behaves as follows:

- An XSL engine converts the XML document into a tree structure, composed of various types of nodes.
- A style sheet transforms the XML document using pattern matching, via XPath language notation, and then applying rules (templates) within the style sheet.
- In the transformed document, the body of the template element replaces the matched node in the source document.

XSLT has a very neat capability to do functional transformations by escaping into another language using Extensible Markup Language Formatting Object (XML-FO). The W3C's XSLT recommendation does not define any aspect of this mechanism nor does it require that an XSLT processor should provide one at all. Future versions of XSLT should remedy this. XML-FO used in the Cocoon Web Publishing Framework provides the following features for XSL-FO:

- Specifications transformed into Portable Document Format (PDF) allowing for an easy integration when rendering PDF documents.

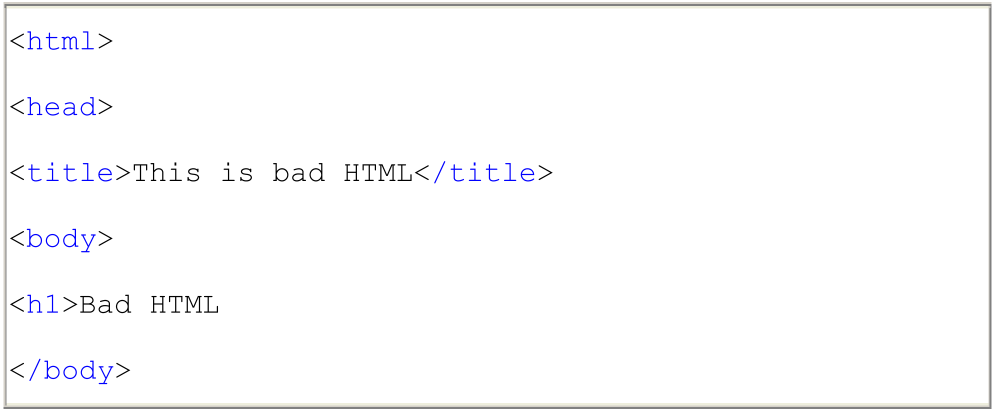
- Specifications generate a Scalable Vector Graphics (SVG) image out of a dynamically generated page. Database graphs and vector counters are also used in Cocoon.
- Specifications when mixed with other graphic outline specification such as SVG and PDF. Both are interpreted by Formatted Object Protocol (FOP) and rendered into PDF as area and vector graphics.
- Generation of VRML by applying the appropriate stylesheet using XML-FO.
- Dynamic ability to understand which browser is requesting the page and applies a different stylesheet to the same XML page to render in different client formats (e.g, Wireless Markup Language (WML),Wireless Access Protocol (WAP), browser/cell-phone, Personal Digital Assistant (PDA)).

Further discussion on extension functions can be found in [Kay00].

F. EXTENSIBLE HYPERTEXT MARK LANGUAGE (XHTML)

XHTML is a combination of HTML and XML (eXtensible Markup Language). XHTML consists of all the elements in HTML 4.01 combined with the syntax of XML. The Web has reached a point where many pages on the WWW contain "bad" HTML (Figure III.6).

The following HTML code will work fine if you view it in a browser, even however does not follow the HTML specification and trying to exstract data required error correcting code.



```
<html>

<head>

<title>This is bad HTML</title>

<body>

<h1>Bad HTML

</body>
```

Figure III.6 Example of bad HTML

XML is a markup language where everything has to be marked up correctly, which results in "well-formed" documents. XML was designed to describe data and HTML was designed to display data. The WEN will consist different browser technologies, some browsers run internet on computers, and some browsers run internet on mobile phones and palm pilots. The last-mentioned do not have the resources or power to interpret a "bad" markup language.

Therefore by combining HTML and XML, and their strengths, W3C producted a markup language that is useful now and in the future. XHTML pages can be read by all XML enabled devices and while waiting for the rest of the world to upgrade to XML supported browsers, XHTML gives you the opportunity to write "well-formed" documents now, that work in all browsers and that are backward browser compatible. XHTML is XML; an XHTML (XML document) is validated against a either a DTD or a XML Schema. The output is parsed against the XHTML XSL stylesheet.

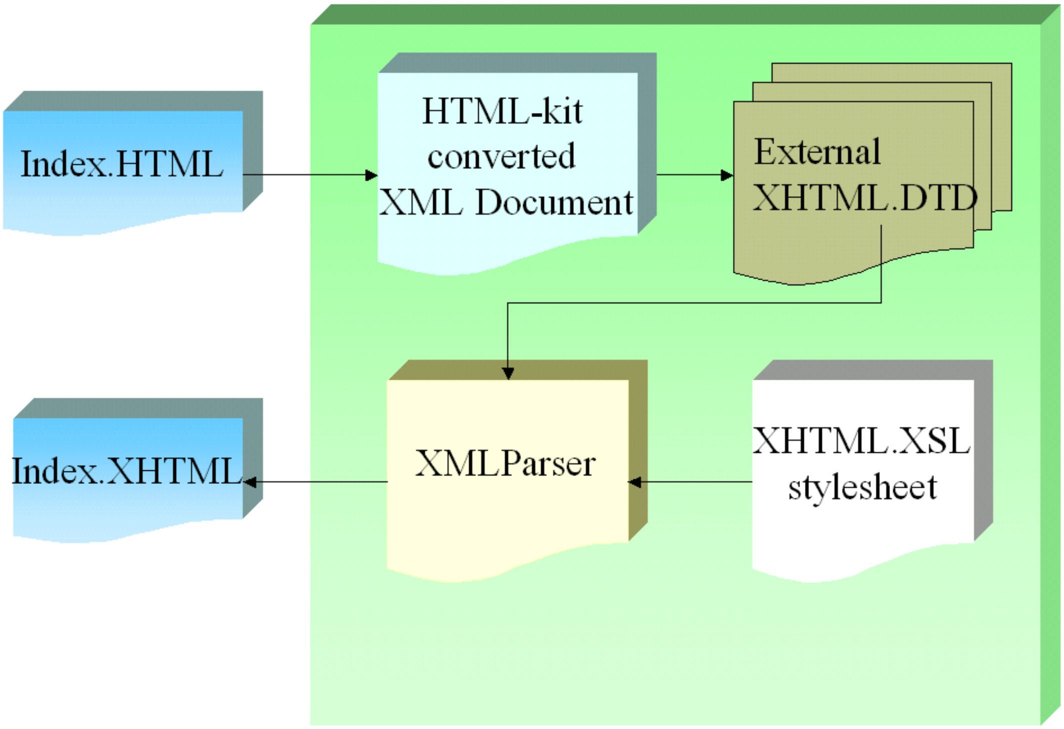


Figure III.7 HTML to XHTML Conversion

The framework prototype site uses XHTML in the Web site is design (Figure III.7). All HTML files are converted into an XML document. A DTD (Appendix C Section E) validates the XML files and applies an XHTML stylesheet to create the output file of XHTML. Example of the converision code is contained in (Appendix C). In the Web publishing framework prototype site, all Web pages are designed using FrontPage and converted to XHTML using HTML-Kit. HTML-Kit is free software available at http://www.chami.com/html-kit/. HTML-Kit validates the HTML and convers the HTML to a well-formed XML document. The HTML-Kit automatically assigns a DTD to validate the XML document, and applies an XHTML stylesheet to create XHTML.

G. XML COMPONENTS

While there are many components and rules for creating a proper XML document, only XML components incorporated in this thesis are discussed. There are countless numbers of XML books, Web pages, magazines, presentations, and other resources available for readers who desire more information on the subject. A good starting point is the World Wide Web Consortium (W3C) home page at http://www.w3.org.

Elements and attributes are the fundamental nodes used to create XML documents. Elements consist of a start and end tag, and are characteristically given descriptive names to denote meaning. In figure III.9 `<SUBMARINE_OFFICER> LCDR Smart </SUBMARINE_OFFICER>` is the element `<SUBMARINE_OFFICER >` containing the data "Sub Off: *LCDR Smart*"(Figure III.8). Tag names must begin with a letter, underscore, or colon. Follow on characters can be letters, numbers, etc., but cannot contain white space. As noted earlier, it is easy to depict the same data using name spaces. Because XML parsers are case sensitive, the element like `<Engineering_Duty_Officer >` is not the same as `<engineering_duty_officer>` or `<ENGINEERING_DUTY_OFFICER>`, but (to humans at least) all three versions clearly describe the same data.

```
<html>
<head>
<title>Command Structure</title>
</head>
<body>

<p>CO: Capt Winitall</p>
<p>XO: CDR Captiwant</p>
<p>Repair Fac: LDCR Misslefix</p>
<p>Sub Off: LDCR Silence</p>
<p>Carrier Off: LCDR Cruiser</p>
<p>Sup Officer: LT Fast</p>

</body>
</html>
```

Figure III.8 HTML Component Example

```
<ROSTER>
  <TITLE>Command Structure</TITLE>

   <COMMANDING_OFFICER>Capt Winitall</COMMANDING_OFFICER>

   <EXECUTIVE_OFFICER>CDR Captiwant</EXECUTIVE_OFFICER>

   <REPAIR_FACILITIES_OFFICER>LDCR Misslefix
</REPAIR_FACILITIES_OFFICER>

   <SUBMARINE_OFFICER>LCDR Smart</SUBMARINE_OFFICER>

   <CARRIER_OFFICER>LCDR Cruiser</CARRIER_OFFICER>

   <SUPPLY_OFFICER>LT Fast</SUPPLY_OFFICER>
</ROSTER>
```

Figure III.9 XML Component Example

Attributes enable an author to define extra information about an element. Values given to attributes are strings that must be delimited by quotation marks. An example of an attribute is <Service branch=”USN”> where <Service> is the element with the attribute branch having the value “USN”. There are arguments for and against the use of attributes. Arguments opposed to the use of attributes include the inability of attributes to specify structure, and their tendency to inhibit document readability when they are used in excess. Arguments in favor of using attributes included usefulness in describing the unique properties of an object and therefore increasing the quality of the metadata provided. In any case, a good rule of thumb is to use an attribute for simple, unstructured “secondary” information that is unique and cannot be separated from the concept of the element. [Nick01]

H. JAVA J2EE TECHNOLOGY

The Web-publishing framework uses Java Technology, which supports the Java 2 Enterprise Edition (J2EE). At the application layer, enterprise wide systems are recommended. Java 2 Enterprise Edition (J2EE) is the preferred distribution object model. J2EE is based upon open standards that promote common interfaces for object use, storage, and run-time interactions. [WEN01] J2EE provides the following capabilities:

- J2EE specifies architecture for developing and deploying distributed applications.
- J2EE promotes the 3-tier distributed software architecture.
- J2EE is built on a server-side component model.
- J2EE promotes the development of interoperable software components.
- J2EE provides the robust and reliable system-level services, which distributed applications, require, such as data access, transaction management, security, and client connectivity.

1. Java Servlet

Java Servlets are more efficient, easier to use, more powerful, more portable safe, and cheaper than traditional CGI and many alternative CGI-like technologies. Servlet technology is very general. The Servlet model is designed to be used over any protocol or with clients other than Web browsers. Currently, the Web-publishing framework uses the Servlet technology for Web applications. For this reason, we review only one special kind of Servlet: HTTP Servlets.

HTTP Servlets extend the HTTP Servlet class, which implements the Servlet interface. These Servlets provide more features than just fast response to clients. For instance, GET parameters that are passed through a Web browser to a Web server are available to the user of the Servlet API. Similarly, output sent from a program is sent back to the browser on the behalf by the Web server Servlet response. Two or more requests for the same Java class run in multiple threads (as opposed to multiple processes in the CGI model). This means that a Java class is thread safe.

Similarly, when requests for a Java class cease, the instance is not extinguished. Instead, in anticipation of future requests, the memory and other resources are reused when necessary. Of course, Java garbage collection takes place as usual.

An HTTP Servlet's job is to read any data sent by the user usually via a Web page. The Servlet looks up information about the request that is embedded in the HTTP request such as cookies, host name, etc. and generates the results. Appendix D shows a simple example of Servlet and JSP technology.

2. Java Server Pages (JSP)

Java Server Pages (JSP) is similar to the dynamic part of Active Server Pages (ASP), except it is written in Java rather than VBScript or another ASP specific language, so it is more powerful and better suited to complex applications. JSP is portable to other operating systems and Web servers, and is a free, open source HTML-embedded scripting language.

JSP does not provide any capabilities that could not be accomplished with a Servlet. JSP are automatically translated into Servlets behind the scenes. JSP improves the dynamic HTML generation by separating the presentation from the content. The Web page design expert can build the HTML using familiar tools and leave places for programmers to insert the dynamic content.

Another reason to have multiple Servlets and/or JSPs communicating is to produce loosely coupled reusable Web components. The JSP example in Appendix D is an input form used to communicate between JSP and a Servlet.

3. Enterprise Java Beans (EJB)

EJBs are a high-level specification for a server side component model. EJBs are designed to run in distributed environments. The beans are built on top of the Remote Method Invocation (RMI), thus EJB support distributed computing. EJBs provide the following:

- Standard component model for distributed applications written in Java.
- An easy way of creating distributed applications by hiding low-level details such as location, transaction processing, object creation, and persistence.
- Platform independence running in any EJB container on any platform.
- Compatibility with existing application server's platforms, and other Java API's allowing vendors to include EJB functionality without disabling support for other technologies.
- Compatibility with COBRA.

An EJB container consists of an EJBHome, EJBObject, and the EJB. The container serves the function of acting as a holder of EJBs. The container resides on the server. It provides security, transaction services, concurrency, data persistence, and other services completely transparent to the beans inside the container and to the users of the beans outside the container. The container contains multiple types of EJB's each with an EJB home interface to allow interactions with the container and the clients. The client application "finds" the container by calling on the JAVA Naming and Directory Interfaces (JNDI) service. Essentially, it is a phonebook for distributed objects, providing information on how to connect to the bean. The EJB handles the establishment of communication between client and server. The client application gets a reference to an EJB through the home interface and communicates to the underlying bean through the EJB object stub class. EJB's architecture provides multiple database connections and offer an easy way of creating a distributed application.

This research did not incorporate the EJB component, which the Web application framework is fully capable of supporting because the goals was to demonstrates and evaluates the functionality of an XML-based Web Publishing Framework. The Web framework prototype in chapter IV demonstrates ability to connect to RBDMS with either a generic Servlet connection which converts non-XML data to XML format or a completely XML-base systems using the Web Publishing Framework to convert non-XML data to XML format. [CoreS].

IV. IMPLEMENTATION OF THE WEB APPLICATION FRAMEWORK TO CONNECT TO HETEROGENEOUS DATABASES.

A. INTRODUCTION

US Navy needs to migrate from centralized database architecture to globally distributed replicated heterogeneous systems. One of the greatest challenges facing today's US Navy is the integration of heterogeneous systems without compromising local autonomy, reliability, and transparency. The distributed database is growing in importance since the current client-sever central architecture provides access to many databases. The US Navy needs to create a system by which uniform storage, access and retrieval is being made possible. Currently Web network services have independent processes for collecting, analyzing, and displaying data. There is no truly interaction or automatic information sharing the elements in the U.S. Navy network. This results in duplication of effort since some commands might collect and analyze the same information without realizing that similar information exists at other sites. Further, due to the independent nature of these sites, the Navy loses the benefit of accessing data on other sites.

A distributed database is single logical database that is spread physically across computers in multiple locations that are connected by a data communications link. Distributed Database system with multiple sites needs to maintain local autonomy to be able to support local DBMS and provide manageable the system by the Database Administrator (DBA). Local Autonomy provides the ability of systems interoperate across multiple operating systems. Autonomy is needed because each system may be specializing in an area and could treat some of the data as irrelevant to its audience.

Distributed DBMS must be able to maintain location transparency, replication transparency, and failure transparency. Location transparency allows users to make queries or update to database systems and not know the location of the data. It appears to the user as a logical database stored at one site. Replication transparency allows the users to access data and not to realize that the data is copy. Also, failure transparency this hides transactions failures for the user. Transaction failures need to support a concurrency. Either all or none of the actions of a transaction is committed using concurrency control scheme to ensure data integrity. Distributed DBMS Transparency Objectives should incorporate a Commit Protocol: Ensures that a global transaction is either successfully completed at each site or else aborted and support asynchronous transactions. An XML-based Web Publishing Framework with the Java technologies provides solutions for distributive database architecture.

Additionally, this section will shows the prototypes generic XML access distribute database architecture and provide a prototype mechanism in figure IV.1 to share XML data across a network using the Web Publishing framework. This architecture uses a uniform environment, which allows for ad hoc networks. In this way, a user familiar to a site can access information from other sites. The design provides users the autonomy in the choosing the operating system, the storage medium, and the user interface[Will00].

There are a number of reasons why the Navy needs to expose the database information as XML listed below:

- Sharing data with other systems
- Interoperability with heterogeneous system
- Exposing legacy data to application that use XML
- Business to business transactions
- Object persistence using XML
- MVC design approach
- Content syndication

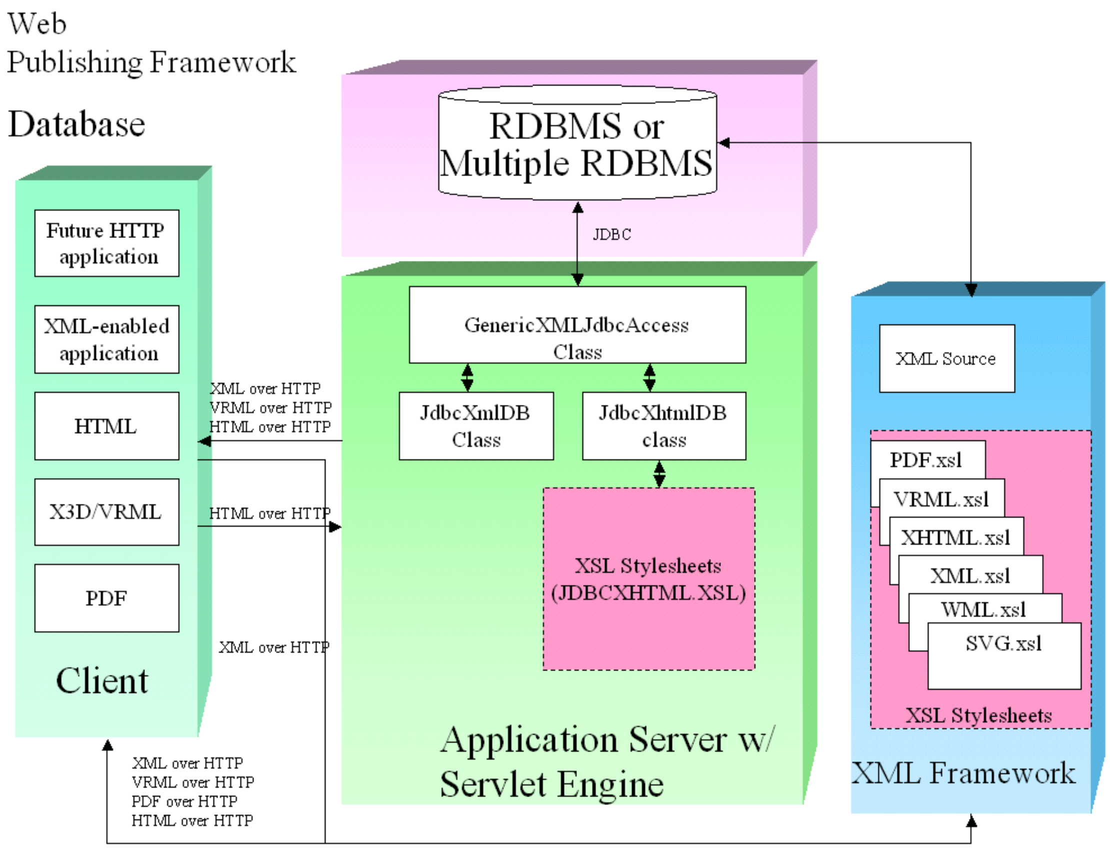


Figure IV.1 Generic Database Access Web Publishing Framework

The prototype demonstrates XML's integration with RDBMS. The research shows how to store XML data in its native format and in a relation database, as well as how to create generic XML database access model using both Java Servlet technology and XML technology that allow quick and efficient access such as data driven Web pages. This section will examine an approach to taking existing relational database and moving it to a fully controlled XML format.

B. ENVIRONMENT

XML provides the user the ability to represent structured data without any additional information. Because this structure is inherent in the XML document rather then needing to be driven by an additional document that describes how the structure appears, it becomes very easy to sent structured information between systems. Since

XML documents are simply text files, they may also be produced and consumed by legacy systems allowing these systems to expose their legacy data in a way that can easily be accessed by different consumers.

Another advantages to the use of XML is the ability to leverage tools, either already available, or starting to appear, that use XML to drive more sophisticated behavior. For Example, XSLT may be used to style XML document producing XHTML documents, X3D, VRML, WML decks or any other type of text documents. XML Web publishing framework allows XML to be encapsulated in routing information, which then may be used to drive documents to the user in desired output.

The framework system implements Java because Java is portable across various operating systems and Storage devices. In addition, Java Servlets are tightly coupled with the Java Object Oriented language and J2EE. With the emergence of the XML as a common information medium, and given the issues of dependability and availability of the Internet, the need for asynchronous distributed transactions has escalated. Oracle is the predominant the database system. However, other relational databases such as SQL 2000, Access, Informix, and Sybase could also be connected using the Web-publishing framework and a JDBC. Similarly, UNIX and Windows are common operating system used in the Navy's network. Thus, the prototype system uses technologies that interoperate across multiple operating systems. The system provides autonomy because XML enable the user at each node to treat some of the data as irrelevant to its audience.

The framework implements a nodal hierarchic structure using XML and JAVA that will provide a quick connection to databases. The structure concept of nodal encapsulates metadata using XML as a common reference.

C. FUNCTIONALITY XML BASED WEB FRAMEWORK

When you combine XML with J2EE technologies and standard Internet protocols used in the Web Publishing Framework the user can quickly develop enterprise applications that provide universal data access to JDBC data sources with a minimal amount of coding. With XML, XSLT, and JDBC the developer can design scalable and extensible architectures that facilitate device independent data delivery in J2EE applications. This allows end user to access your data using Web browsers or any other current of future devices that render content using an XML based Markup Language. Combining XML, Java, and JDBC facilitated the creation of Web services using J2EE.

The JAVA Servlet and XML database connection in appendix F. uses JDBC Data Source and Driver. The JDBC Data Source and Driver support a variety of different relational database products. Most database products on the market today include a JDBC driver (Oracle, Microsoft SQL, MySQL and Microsoft Access). The fully implemented XML database connection in appendix G. uses JBDC connection within a XML Schema. XML Schema also provides a mechanism for capturing information used to establish correspondence between systems. The schema defines the structure, syntax, and to some extent, the semantics of the component system external interface.

D. EXAMPLE OF DATABASE CONNECTIVITY USING THE WEB FRAMEWORK WITH SERVLETS

This section discusses the JAVA Servlet and XML database generic connection architecture (Figure IV.2). The main control section for heterogeneous database access using JDBC is the class GenericXMLJdbcAccess. This class controls access to databases using Servlets, executes SQL statements using the JDBC data source, serializes the returned result set in XML format, and returns an XML document to the requestor. In the architecture, the GenericXMLJdbcAccess is reused by two Java Classes: JdbcXmlDB and JdbcXhtmlDB. The JdbcXmlDB provides a generic XML over HTTP interface to JDBC data source for XML-enabled applications. The JdbcXhtmlDB returns a result set as HTML over HTTP to the Web browser using the specified XSL stylesheet. The system in figure IV.2. essentially acts as a request controller for the database connection, which delegates handling to the JdbcXhtmlDB or JdbcXmlDB classes.

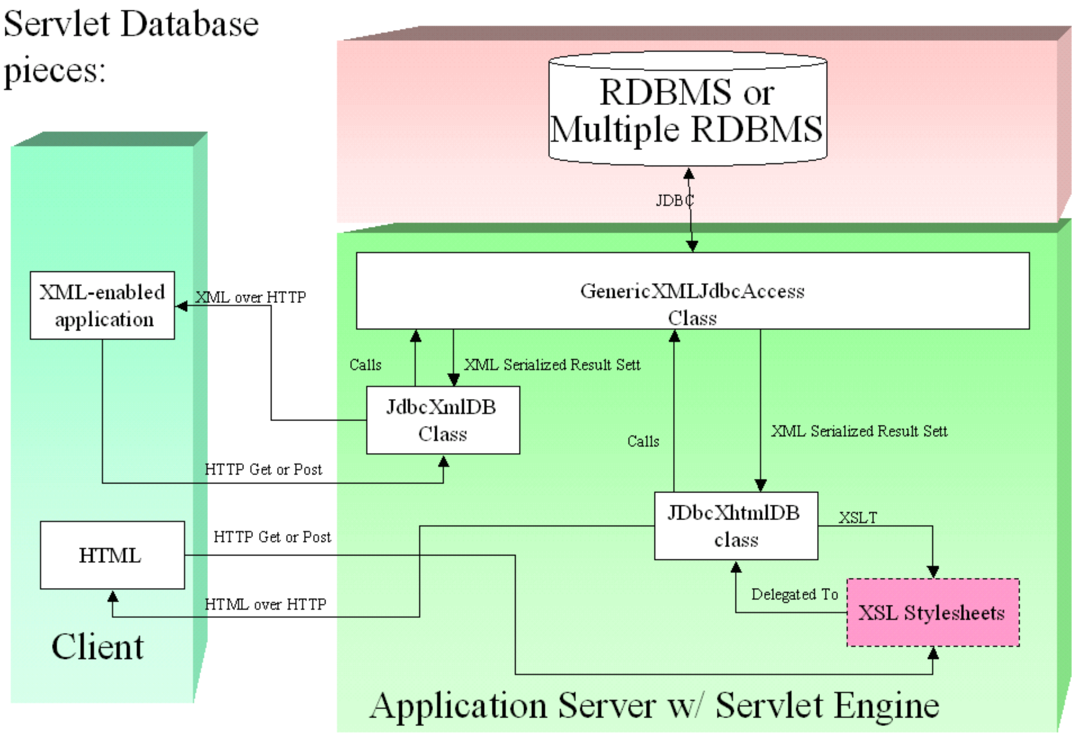


Figure IV.2 JAVA Servlet and XML database generic connection architecture

1. GenericXMLJdbcAccess Class

The GenericXMLJdbcAccess class encapsulates the functionality required to query a database and return the results of the query to develop a well-formed XML document (see Appendix E, Section B for the associated source code). The encodeXML() method applies the XML encoding rules for generic data access. The method simply does a “search and replace” operation on the string into the method, replacing instances of special characters. The method will be used by writeXML() method to ensure that any data retrieved form the database is encoded properly before serialization to XML. The writeXML() method creates a DTD-less document, which is parsed by a non-validating XML parser, and returns serialized string into an XML document. The XML document contains the result set, that is metadata and data values, as seen in Appendix E, Section E. The execute() method does the following:

```
/**
 *
 *  Establish a connection to a database and executes the SQL
Query returns String Resultset
 *   formatted in XML
 *
 */
 public String execute(String driver, String url, String uid,
String pwd, String sql)
 {
 String output = new String();
 try
 {
    Class.forName(driver);
    Connection conn =
DriverManager.getConnection(url,uid,pwd);
    Statement s = conn.createStatement();
    ResultSet rs= s.executeQuery(sql);
    output = writeXML(rs);
    rs.close();
    conn.close();
 }
 catch(Exception e)
 {
    output = "<error>" + encodeXML(e.toString()) + "</error>";
 }
 return output;
 }
```

Figure IV.3 GenericXMLJdbcAccess.Class Execute( ) Method

- sql - This argument contains the complete text of an arbitrary SQL92 command.
- s.executeQuery - This triggers the execution of the SQL command.
- rs is the result- this event fires when the SQL command completes executing. It contains a numeric error code, or zero for success.
- Exception handles the errorMessage –This event fires when the SQL command completes execution. It contains a text error message, or an empty string for success.
- url, UID, PSD - This field references a database URL, user identification, and password that enables access to the database.
- writeXML(rs)- passes the result set to write for conversion, and creates the XML serialized output.

2. JdbcXmlDB Class

The JdbcXmlDB class provides an XML over HTTP interface to a JDBC database (see Appendix E, Section C for the associated source code). The Servlet implements the javax.Servlet.http.httpServlet.class and JdbcXmlDB uses the doGet() and doPost() methods from the httpServlet.class. The doGet() method instantiates the GenericXMLJdbcAccess class and calls the execute() method. The doPost() method overrides the doGet() method by simplifying the call to doGet(). This method returns the XML serialized result set to respond to each HTTP request.

3. JdbcXhtmlDB Class

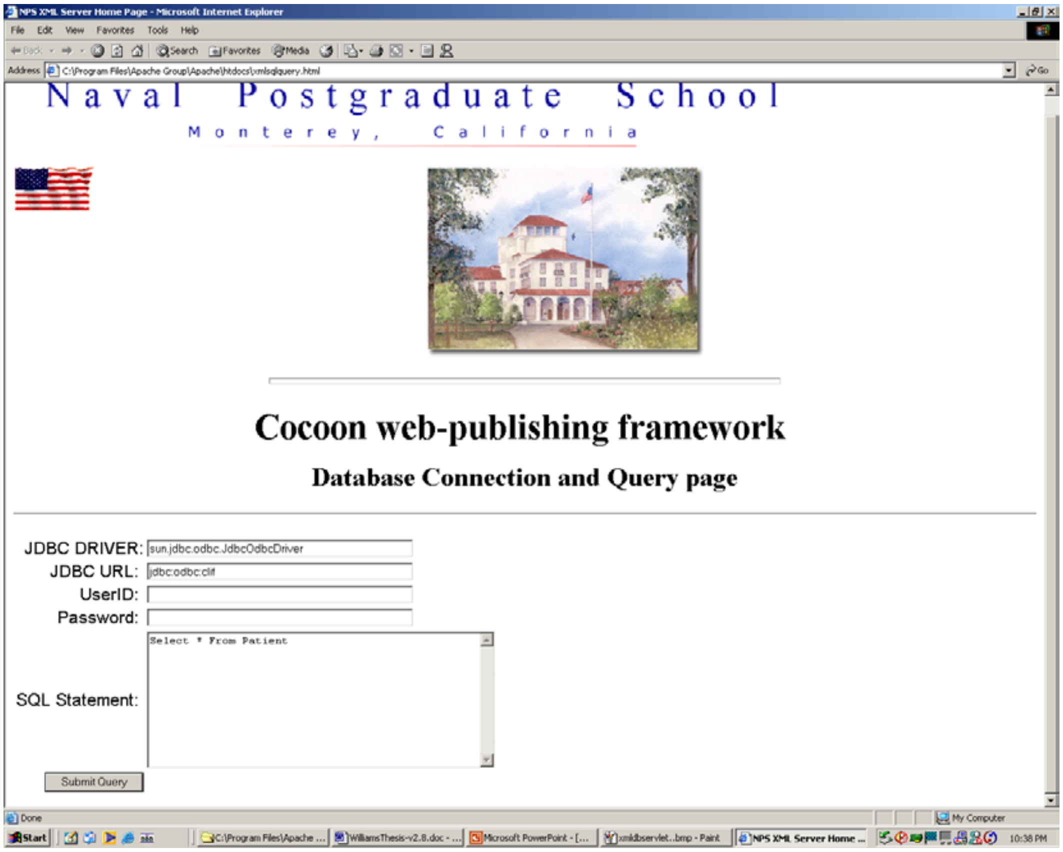
The JdbcXhtmlDB class provides an HTML over HTTP interface to a JDBC database (see Appendix E, Section D for the associated source code). The Servlet implanting the javax.Servlet.http.httpServlet.class. This class also uses the doGet() and doPost() methods. The doGet() method instantiates the GenericXMLJdbcAccess class and calls the execute() method. The doPost() method overrides the doGet() method by simplifying the call to doGet(). The GenericXMLJdbcAccess returns the XML serialized result set document to this Servlet. The Servlet then makes a call to the DOM method (figure IV.5.) using "XSLTProcessor proc = XSLTProcessorFactory.getProcessor();" and the "proc.process(**new** XSLTInputSource(**new** java.io.StringReader(output)), **new** XSLTInputSource(xsltSystemId), **new** XSLTResultTarget(xmlout));" functions. The DOM parses the "output" result set XML document with the "xsltSystemID" XSL Stylesheet (Appendix E, Section F). This method returns an HTML serialized result set "xmlout" to respond to the HTTP request (figure IV.5.). Appendix E, Section I shows the HTML result from the JdbcXhtmlDB, which applies the JDBCHTML.xsl stylesheet.

```
Public class JdbcXhtmlDB extends HttpServlet
{

  public void doGet(HttpServletRequest req, HttpServletResponse res)
     throws IOException, ServletException
  {
  String qryPath;
  String xsltSystemId = new File("C:/jakarta-tomcat-
3.2.3/Webapps/jdbcxml/WEB-
INF/classes/jdbc/xml/JDBCHTML.xsl").toURL().toExternalForm();


  if(req.getPathInfo()==null)
  {
     qryPath =
getServletConfig().getServletContext().getRealPath(req.getServletPath());

  }
  else
  {
  qryPath = req.getPathTranslated();
  }
  res.setContentType("text/html");
  PrintWriter xmlout = res.getWriter();
  GenericXMLJdbcAccess searchOBJ = new GenericXMLJdbcAccess ();
  String output = searchOBJ.execute(req.getParameter("driver"),
     req.getParameter("jdbcurl"), req.getParameter("uid"),
     req.getParameter("pwd"),req.getParameter("sql"));
  try
  {
  XSLTProcessor proc = XSLTProcessorFactory.getProcessor();
  proc.process(new XSLTInputSource(new java.io.StringReader(output)),
     new XSLTInputSource(xsltSystemId), new XSLTResultTarget(xmlout));
  }
  catch(SAXException se)
  {
     throw new ServletException(se);
  }
 }
```

Figure IV.4   JdbcXhtmlDB.class doGet( ) Method

4. XHTML Interface Form

Figure IV.5. shows the XHTML interface form (see Appendix E, Section G for the associated XHTML code). The interface form was developed in HTML and converted into an XML document. A DTD validated the XML files and applied an XHTML stylesheet to create the output file of XHTML. The XHTML file uses user, database connection and an SQL query statement to call JdbcXhtmlDB. Appendix E, Section F shows the XHTML XML document.


NPS XML Server Home Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address C:\Program Files\Apache Group\Apache\htdocs\xmlsqlquery.html
Go
Naval Postgraduate School
Monterey, California
Done
My Computer
Start
C:\Program Files\Apache ...
WilliamsThesis-v2.8.doc - ...
Microsoft PowerPoint - [...
xmldbservlet...bmp - Paint
NPS XML Server Home ...
10:38 PM

Cocoon web-publishing framework

Database Connection and Query page

JDBC DRIVER: sun.jdbc.odbc.JdbcOdbcDriver

JDBC URL: jdbc:odbc:clif

UserID:

Password:

SQL Statement: Select * From Patient

Submit Query

Figure IV.5 XHTML interface form

E. EXAMPLE OF DATABASE CONNECTIVITY USING THE WEB FRAMEWORK WITH SERVLETS AND XML

The XML-Based Web Publishing Framework allows for database connection using ESQL (Figure IV.6). ESQL is a very new technology, evidenced by the sparse amount of documentation currently available. This section documents a generic database connection using XML-based Web Publishing Framework. This design can support multiple queries to different databases. The connections and queries can be built in an XML document GenericESQL.XML (Figure IV.7). The XML document is validated against a schema using the ESQL.XSD document (Appendix F, Section B). The Schema is a formal specification, written in the W3C standard language XML Schema, of every single ESQL element and attribute. The documents are human readable and include comments for the documentation of each tag. The XML document is used by the framework to connect to the database using the JDBC:ODBC Bridge in the XML database generic connection architecture (Figure IV.6). The framework automatically responds to client base to XML document request by using the XML document extention and document name selected by the user("helloworld.xml").

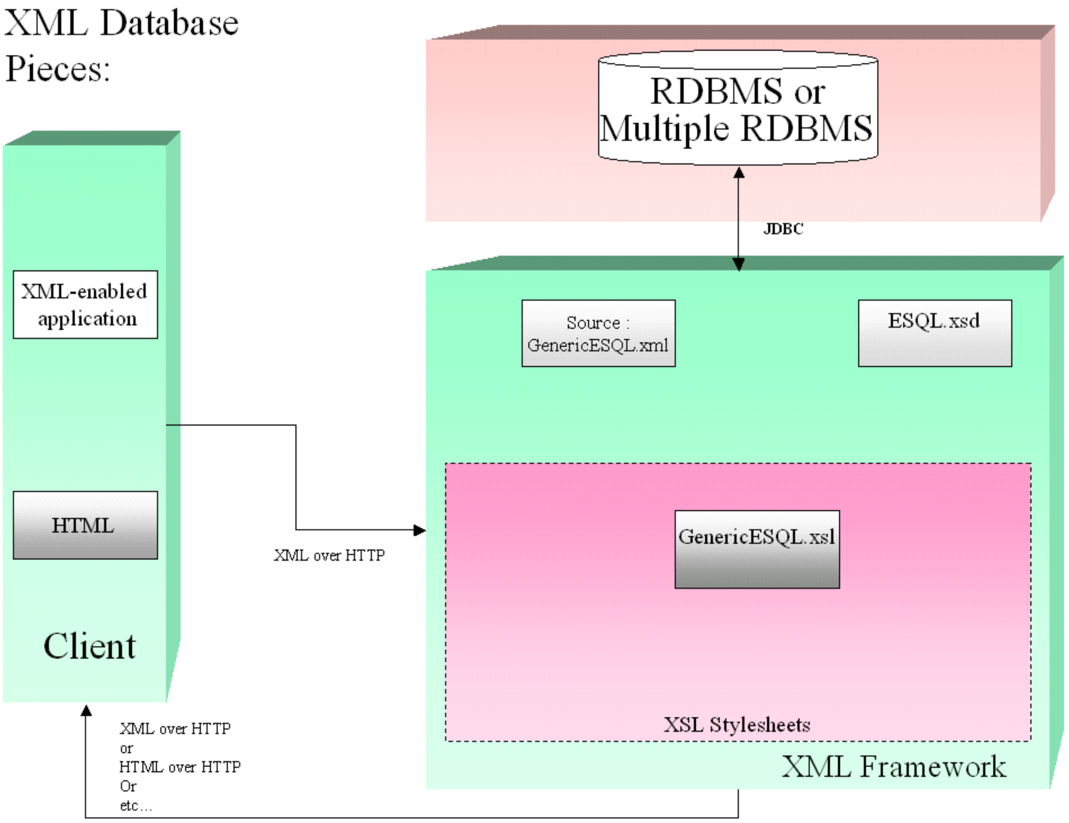


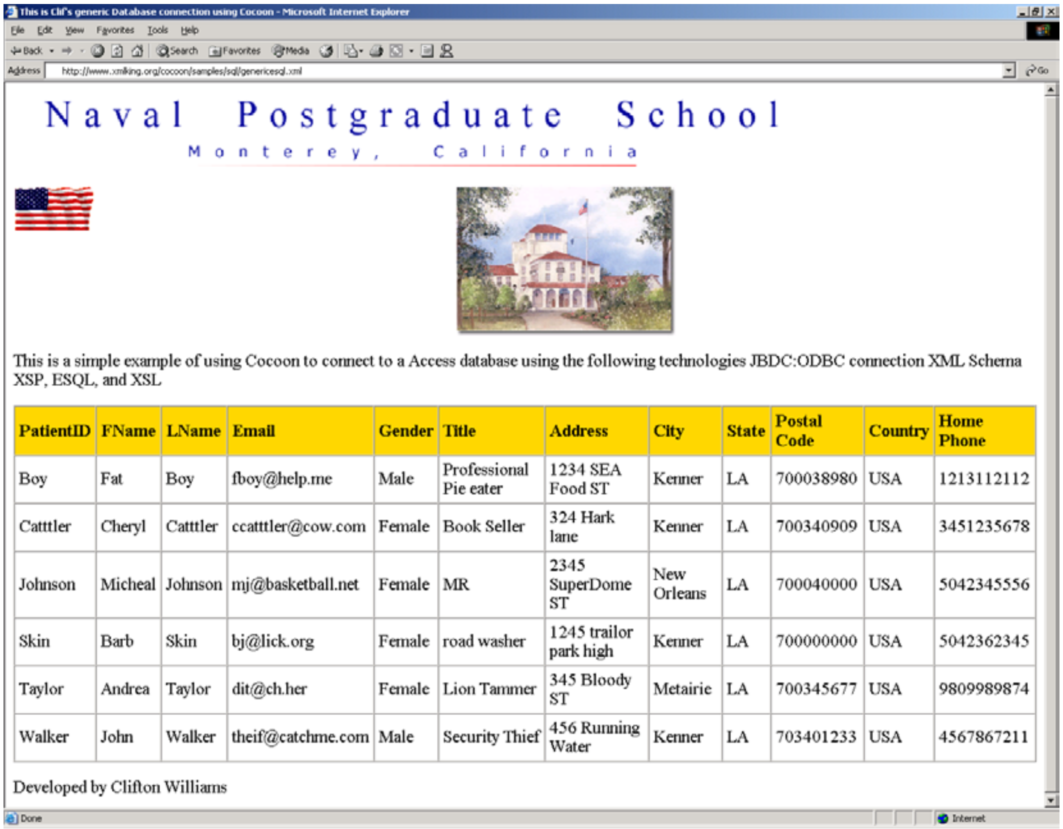
Figure IV.6 XML database generic connection architecture

The GenericESQL.XML (Figure IV.7) document is requested by the client browser via a call to the XML document using the Web Publishing Framework URL and document name. The framework, in response to the users request, calls several processes using XSP and ESQL namespaces and converts the XML data using the GenericESQL.XSL stylesheet (Appendix F, Section C). The XML document is validated against the ESQL Schema (Appendix F Section B). The XML document is the main control section for heterogeneous database access using JDBC via the GenericESQL.XML (Figure IV.7). This document controls access to databases using XSP commands based upon the connection tag set. The document also executes SQL statements using the JDBC data source, serializes the returned results set in XML format, and returns an XML document to the requestor. The XML document also controls which output is applied. The output could be XML over HTTP to a JDBC data source for XML-enabled applications, or HTML over HTTP to a Web browser (Figure IV.8) using the specified XSL stylesheet with the document.

```
<?xml version="1.0"?>
<!-- edited with XML Spy v4.0 U (http://www.xmlspy.com) by Clifton Williams (Naval
Postgraduate School) -->
<?cocoon-process type="xsp"?>
<?cocoon-process type="xslt"?>
<?xml-stylesheet type="text/xsl" href="genericesql.xsl"?>
<xsp:page xmlns:xsp="http://www.apache.org/1999/XSP/Core"
xmlns:esql="http://apache.org/cocoon/SQL/v2">
   <page>
      <esql:connection>
         <esql:driver>sun.jdbc.odbc.JdbcOdbcDriver</esql:driver>
         <esql:dburl>jdbc:odbc:clif</esql:dburl>
         <esql:username/>
         <esql:password/>
         <esql:execute-query>
            <esql:query>Select PatientID, FName, LName, Email, gender, Address,
Title, City, State, PostalCode, Country, HomePhone From Patient</esql:query>
            <esql:results>
               <header>This is a simple example of using Cocoon to connect to an
         Access database using the following technologies JBDC:ODBC connection
XML Schema XSP, ESQL, and XSL</header>
               <esql:row-results>
                  <department>
```

```
                            <PatientID>
                                <esql:get-string column="PatientID"/>
                            </PatientID>
                            <FName>
                                <esql:get-string column="FName"/>
                            </FName>
                            <LName>
                                <esql:get-string column="LName"/>
                            </LName>
                            <Email>
                                <esql:get-string column="Email"/>
                            </Email>
                            <gender>
                                <esql:get-string column="gender"/>
                            </gender>
                            <Title>
                                <esql:get-string column="Title"/>
                            </Title>
                            <Address>
                                <esql:get-string column="Address"/>
                            </Address>
                            <City>
                                <esql:get-string column="City"/>
                            </City>
                            <State>
                                <esql:get-string column="State"/>
                            </State>
                            <PostalCode>
                                <esql:get-string column="PostalCode"/>
                            </PostalCode>
                            <Country>
                                <esql:get-string column="Country"/>
                            </Country>
                            <HomePhone>
                                <esql:get-string column="HomePhone"/>
                            </HomePhone>
                        </department>
                    </esql:row-results>
                    <footer>Developed by Clifton Williams</footer>
                </esql:results>
            </esql:execute-query>
        </esql:connection>
    </page>
</xsp:page>
```

Figure IV.7   GenericESQL.XML Document


This is Clif's generic Database connection using Cocoon - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www.xmlking.org/cocoon/samples/sql/genericesql.xml
Go
Done
Internet

Naval Postgraduate School

Monterey, California

This is a simple example of using Cocoon to connect to a Access database using the following technologies JBDC:ODBC connection XML Schema XSP, ESQL, and XSL

PatientID	FName	LName	Email	Gender	Title	Address	City	State	Postal Code	Country	Home Phone
Boy	Fat	Boy	fboy@help.me	Male	Professional Pie eater	1234 SEA Food ST	Kenner	LA	700038980	USA	1213112112
Catttler	Cheryl	Catttler	ccatttler@cow.com	Female	Book Seller	324 Hark lane	Kenner	LA	700340909	USA	3451235678
Johnson	Micheal	Johnson	mj@basketball.net	Female	MR	2345 SuperDome ST	New Orleans	LA	700040000	USA	5042345556
Skin	Barb	Skin	bj@lick.org	Female	road washer	1245 trailor park high	Kenner	LA	700000000	USA	5042362345
Taylor	Andrea	Taylor	dit@ch.her	Female	Lion Tammer	345 Bloody ST	Metairie	LA	700345677	USA	9809989874
Walker	John	Walker	theif@catchme.com	Male	Security Thief	456 Running Water	Kenner	LA	703401233	USA	4567867211

Developed by Clifton Williams

Figure IV.8 GenericESQL HTML over HTTP Browser Display

The system in Figure IV.6 uses an XML document, XSP and Java technology to perform generic database connection queries and generate the resultant display. In this design, all the Java development is isolated from the XML developer. The XML developer using the XML-based Web application framework, can easily establish multiple database queries within the XML document without understanding Java programming behind the scenes, and can produce XML documents in whatever desired format for presentation.

F. ANALYSIS OF RESULTS

The XML-based Web Publishing Framework supports the distributed database architecture using XML as the interface language. A distributed database system with multiple sites needs to maintain local autonomy to be able to support local DBMS requirements for manageability. Using the XML technologies, XML documents with meta-data encapsulation, XML schemas and XML namespaces allows the local DBA to define which information needs to be used within the local DBMS. Distributed DBMS must be able to maintain location transparency, replication transparency, and failure transparency. XML-based Web Publishing Framework provides location transparency which allows users to make queries or updates to database systems without knowing the location of the data. Failure transparency is built in with the use of Java technologies. J2EE EJBs provide a high-level specification for a server side component model. EJBs are designed to run in distributed environments, so that the beans are built on top of the Remote Method Invocation (RMI). The EJB distributed applications hide low-level details such as location, transaction processing, object creation, and persistence. Platform independence is maintained since any EJB container can run on any platform, thus this architecture supports the heterogeneous solution; further it is compatible with COBRA. Using EJB in XML Based Web Publishing Framework is just a matter of setting the Servlet environment to acts a a client EJB. So on one side EJB server running and listening for upcoming request, and on the other side the it starts the Servlet container with the right EJB client libraries so that whatever Servlet (or Cocoon code through XSP page for instance) the user writes can contain EJB invocations. The internet transaction require users to employ a two-phase commit transaction for asynchronous transactions. Asynchronous transactions support the Internet communication medium, and account for the issues of bandwidth, availability, latency and asynchronous transfer of information. In addition, support is provided by distributed composite transactions composed of sub-transactions. Asynchronous transactions allow a unidirectional update to propagate from data source to target. XML documents work well with asynchronous replication in a distributed system, and with occurrences of network failures. XML supports the unlimited demands of data replication. XML reduces the complexity and cost of updating both the logic and the content of a Web site because of the use of the MVC design approach. Whenever a relation is updated, it must be updated at each site that holds a copy. Synchronized updating in near real time requires careful coordination.

The thesis prototype provides insights into addressing the problem of database heterogeneity in a distributed internet-based environment using the XML. The creation of XML wrappers is necessary to make XML the universal mechanism for addressing heterogeneity. This research used wrappers to encapsulate the meta-data information. While XML provides a presentation mechanism for the query results from heterogeneous sources, the reverse process, that is, writing back to heterogeneous sources is also supported. XML provides a significant degree of transparency to underlying data.

The XML-based Web-publishing framework directly addresses challenges facing today's distributed database systems in the form of interoperability, autonomy and interactiveness. The use of Java technologies gives our implementation the flexibility to run on various platforms while the usage of the XML schema to encapsulate data provides autonomy from the underlying DBMS.

Performance issues involving middleware technologies were not addressed in this thesis. The database implementation runs every client's request on its own thread space on the server. This enhances the overall performance of the system in terms of both speed and scalability. The latest versions of Internet Explorer and Netscape were used for the client interface.

V. IMPLEMENTATION OF THE WEB APPLICATION FRAMEWORK TO ENABLE TRANSLATION BETWEEN XML AND VRML 3D SIMULATION.

A. INTRODUCTION

The Virtual Reality Modeling Language (VRML) is a Web-based graphics language for building 3D models. The XML-based Web Publishing Framework easily supports three-dimensional simulations utilizing X3D and VRML. Three-dimensional (3D) modeling and simulations provide insights into operations, which cannot be unrealized using standard two-dimensional (2D) renditions. The XML-based Web Publishing Framework supports three-dimensional modeling and simulations for operational planning and provides the U.S. Navy a better way of realizing operational limits.

VRML allows users to interaction within a scene through viewpoint, movement, and rotation. VRML worlds are created using scene graphs that are comprised of nodes. The nodes are responsible for displaying shapes, interaction, and movement through the virtual world. GeoVRML 1.0 is an extension of VRML that allows for accurate rendering of geographic data. Extensible 3D (X3D) is the next-generation specification for VRML. It was designed to overcome some of the limitations of the original VRML specification and incorporates the use of the Extensible Markup Language (XML). These technologies are scalable within the XML-based Web Publishing Framework architecture (Figure V.1). This approach to networked 3D graphics that is suitable for the generation and distribution of large-scale virtual environments.

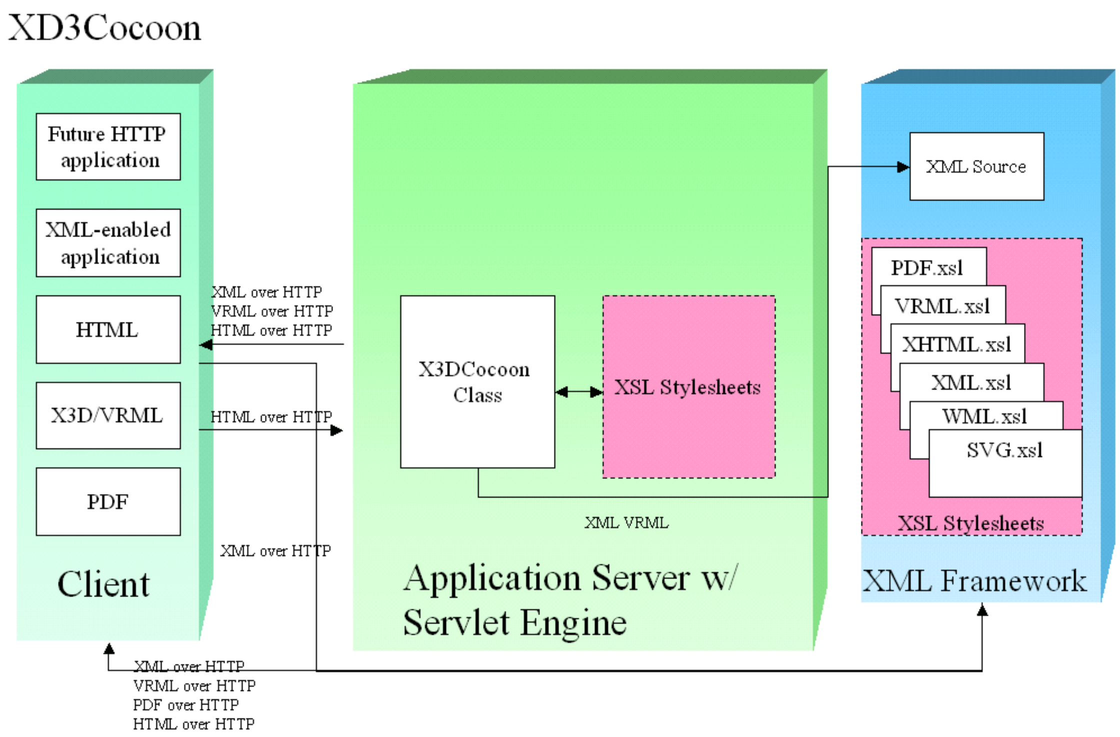


Figure V.1 XML-based Web Publishing Framework for X3D/VRML

The XML-based Web Publishing Framework utilizes an extremely powerful function of XML of applying multiple or chained XSL stylesheets. The XSL stylesheet converts from one XML representation to XML another. This process may involve intermediate steps converting from component system XML to common representation XML.

B. ENVIRONMENT

This section describes concepts that are fundamental for understanding how complex three-dimensional (3D) environments are created. It discusses the Virtual Reality Modeling Language (VRML), DIS-Java-VRML and Extensible 3D (X3D).

1. Virtual Reality Modeling Language (VRML)

The Virtual Reality Modeling Language (VRML) is a Web-based graphics language for building 3D models. VRML allows user interaction within a scene through viewpoint, movement, and rotation. One of the key features of VRML is it is an ISO standard designed for use over the World Wide Web [Nick01].

VRML worlds are created using a scene-graph structure. Scene graphs are simply a hierarchical decomposition of components that will be rendered in a scene. Scene graphs are comprised of various groups of nodes, which together form a virtual world. These nodes are responsible for displaying shapes, interaction, and movement through the world. VRML worlds can be viewed with any VRML-capable browser such as Cosmo Player or Cortona, which can be downloaded at no cost from the internet and support both Internet Explorer and Netscape as plug-ins [Nick01].

There are numerous examples of 3D images and objects available on the Web. These examples demonstrate how groups of nodes are used to create complex 3D objects and virtual worlds. The following list is an excellent reference for 3D examples:

- Naval Postgraduate School VRML Course examples are available at http://www.web3d.org/TaskGroups/x3d/translation/examples/contents.html
- VRML Sourcebook examples at http://www.web3d.org/TaskGroups/x3d/translation/examples/Vrml2.0Sourcebook/contents.html
- VRML/X3D Conformance examples at http://www.web3d.org/TaskGroups/x3d/translation/examples/Conformance/contents.html
- Scenario Authoring and Visualization for Advanced Graphical Environments (Savage) project at http://web.nps.navy.mil/~brutzman/Savage/contents.html

2. DIS-Java-VRML

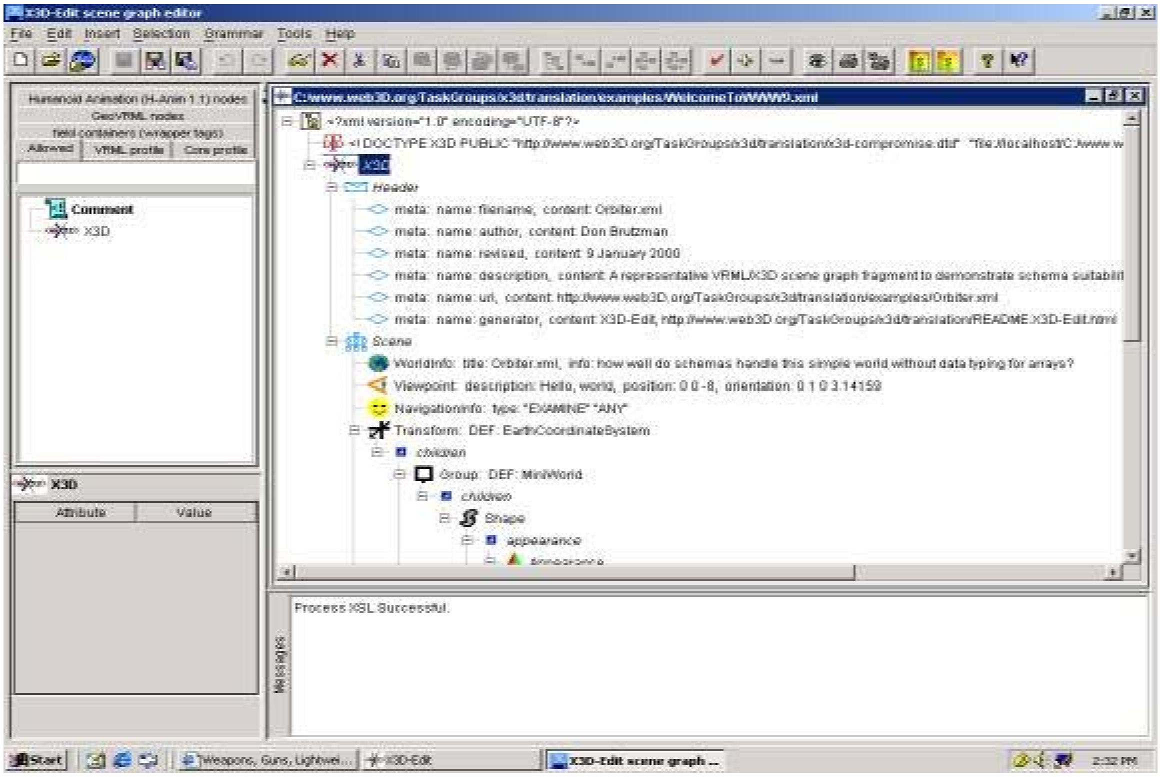
The DIS-Java-VRML Working Group is developing a free open-source software library that allows DIS and VRML to interoperate by implementing Java code. Its goal is to provide libraries and examples that allow scene developers to build networked 3D worlds that are portable, pervasive, scalable and viewable on any computer with a browser and network connection. [Brut01]

3. Extensible 3D (X3D)

The next-generation specification for VRML is the Extensible 3D (X3D) standard [Brut01]. X3D is a scene graph and text-based encoding designed to overcome several limitations of the VRML standard. X3D includes an Extensible Markup Language (XML) encoding to express identical VRML geometry and behavior structures. X3D is thus a standardized XML tagset for describing the VRML 200x standard for Web-compatible 3D content. Such content is not static but dynamic, driven by a rich set of interpolators, sensor nodes, scripts, and behaviors. [Blais01]

X3D/VRML 200x is compatibly with VRML 97 files. X3D provides new interoperability with other relevant standards including MPEG-4 and an entire family of XML-based languages. X3D further addresses several shortcomings of VRML 97, provides tighter media integration, improved visual quality through advanced-rendering nodes, and enables a component-based approach. [Blais01]

Using the X3D-Edit authoring tool, developers can create valid scene graphs with little trouble. The tool is simple enough that even novice 3D developers can create virtual worlds with relative ease. When generating a scene, the X3D-Edit software limits the developer to only see allowable choices. Only valid nodes can be selected, eliminating guesswork and potential errors. X3D-Edit utilizes IBM's Xeena XML editor configured to work with the X3D document type definition (DTD). Documents are translated into VRML using the Extensible Stylesheet Language (XSL), and a browser is automatically launched for convenient debugging. [Brut01] Figure 5.2 shows a screen shot of the X3D Edit tool.

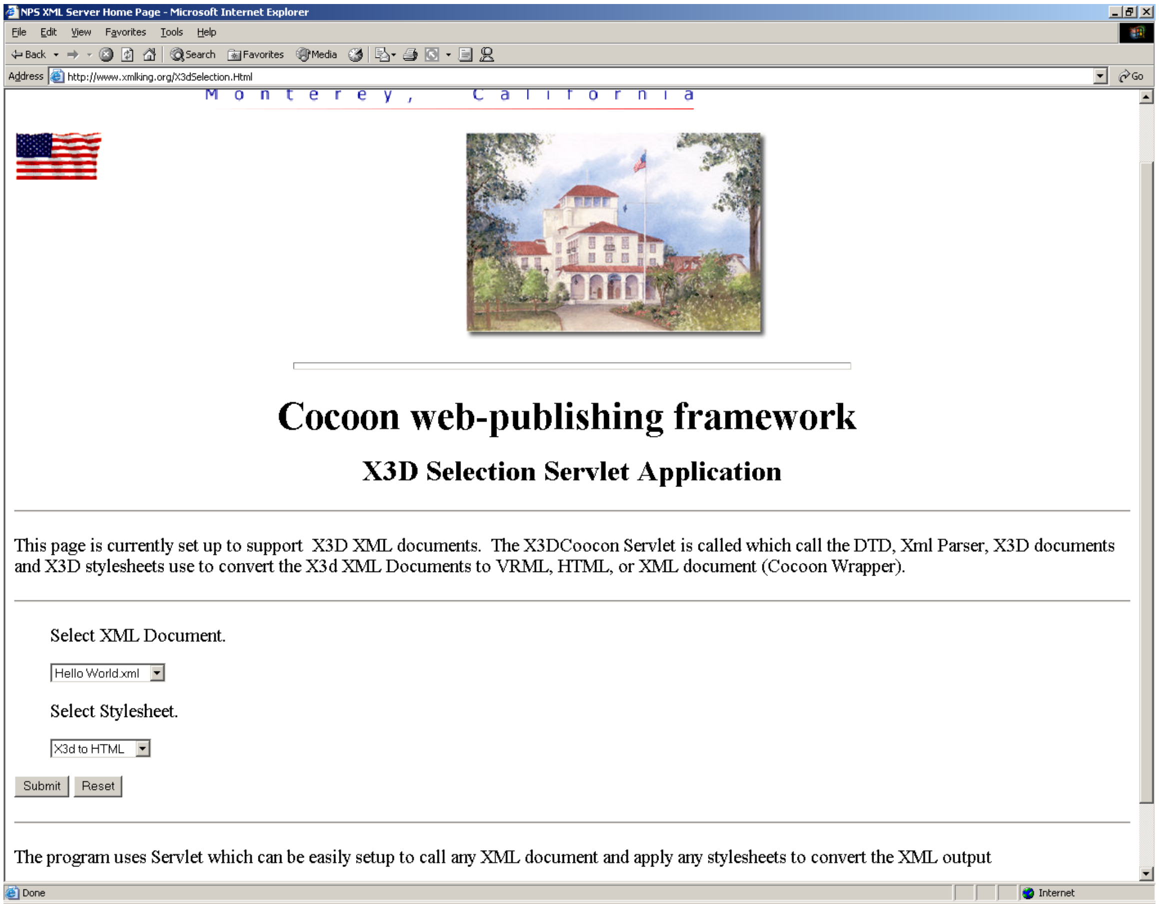

X3D-Edit scene graph editor
File Edit Insert Selection Grammar Tools Help
Humanoid Animation (H-Anim 1.1) nodes
GeoVRML nodes
field containers (wrapper tags)
Allowed
VRML profile
Core profile
Comment
X3D
X3D
Attribute
Value
Messages
Start
Weapons, Guns, Lightwei...
X3D-Edit
X3D-Edit scene graph ...
2:32 PM

```
C:\www.web3D.org\TaskGroups\x3d\translation\examples\WelcomeToWWW9.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "http://www.web3D.org/TaskGroups/x3d/translation/x3d-compromise.dtd"  "file://localhost/C:/www.w
X3D
  Header
    meta: name: filename, content: Orbiter.xml
    meta: name: author, content: Don Brutzman
    meta: name: revised, content: 9 January 2000
    meta: name: description, content: A representative VRML/X3D scene graph fragment to demonstrate schema suitabili
    meta: name: url, content: http://www.web3D.org/TaskGroups/x3d/translation/examples/Orbiter.xml
    meta: name: generator, content: X3D-Edit, http://www.web3D.org/TaskGroups/x3d/translation/README.X3D-Edit.html
  Scene
    WorldInfo: title: Orbiter.xml, info: how well do schemas handle this simple world without data typing for arrays?
    Viewpoint: description: Hello, world, position: 0 0 -8, orientation: 0 1 0 3.14159
    NavigationInfo: type: "EXAMINE" "ANY"
    Transform: DEF: EarthCoordinateSystem
      children
        Group: DEF: MiniWorld
          children
            Shape
              appearance
                Appearance
```

```
Process XSL Successful.
```

Figure V.2 Screen Capture of the X3D Edit Tool

The XML-based Web Publishing Framework utilizing the X3DCocoon Servlet (Appendix G Section B) developed for this prototype system. The Servlet simplifies the rendering of the Web-based graphics language and cuts out the need to store large VRML graphic and HTML files on the server. The Servlet automatically render the VRML, Text, and HTML as requested by the user using only the XD3.xml file and the input XML file, such as HelloWorld.xml (Appendix G Section D). The X3DCocoocn Servlet allows users interaction within the Web Publishing Framework using the X3DCocoon Selection Servlet XHTML Input Form (Figure V.3 and source code Appendix G. Section I). The user can select the X3D XML document and select the X3D XSL stylesheet that the user wishes to apply. The current stylesheet selections available to the users are X3DtoHTML.XSL (Appendix G Section C), X3DtoVRML97.XSL (Appendix G Section J) and X3DCocoon.XSL. The X3DCocoon.XSL stylesheet selection applies a wrapper tagset around the X3DtoVrml97.XSL(Appendix G Section J) output and store the document in at the desired location within the framework.


NPS XML Server Home Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www.xmlking.org/X3dSelection.Html
Go
Done
Internet

Monterey, California

Cocoon web-publishing framework

X3D Selection Servlet Application

This page is currently set up to support X3D XML documents. The X3DCoocon Servlet is called which call the DTD, Xml Parser, X3D documents and X3D stylesheets use to convert the X3d XML Documents to VRML, HTML, or XML document (Cocoon Wrapper).

Select XML Document.

Hello World.xml

Select Stylesheet.

X3d to HTML

Submit Reset

The program uses Servlet which can be easily setup to call any XML document and apply any stylesheets to convert the XML output

Figure V.3 X3DCocoon Selection Servlet XHTML Input Form

C. LIMITATION WEB FRAMEWORK XML-FO FOR X3D AND A SERVLET FIX TO RENDER X3D DYNAMICALLY

Generating dynamic HTML from XML, viewing XML transformed into Text files, and even seeing VRML applications generated from XML is simply a matter of typing the URL to the desired XML file into your browser and watching the XML-based Web Publishing Framework take action. Java libraries are used for converting XML to some non-markup-based formats. The most stable library is the Apache XML group's Formatting Objects Processor (FOP). This gives the XML-based Web Publishing Framework the ability to turn XML documents into VRML simulations.

The Framework uses the XML extensions on the file to automatically pass processing calls to the DOM and can apply a XSL stylesheet to constructs a VRML document. X3D supports the direct translation to VRML; however the XML-based Web Publishing Framework design required users to hand code a software wrapper around the VRML document. The problem encountered is caused by FOP. FOP needs a XML document to process and call the DOM. The VRML document is not a XML document and the XML-based Web Publishing Framework cannot render the VRML file directly. The X3DCocoon Servlet (Figure V.4) corrects these problems by automatically adding a software wrapper around the X3DtoVRML97.XSL output to create a XML document and saves the XML document in the correct directory location for the XML-based Web Publishing Framework to render.

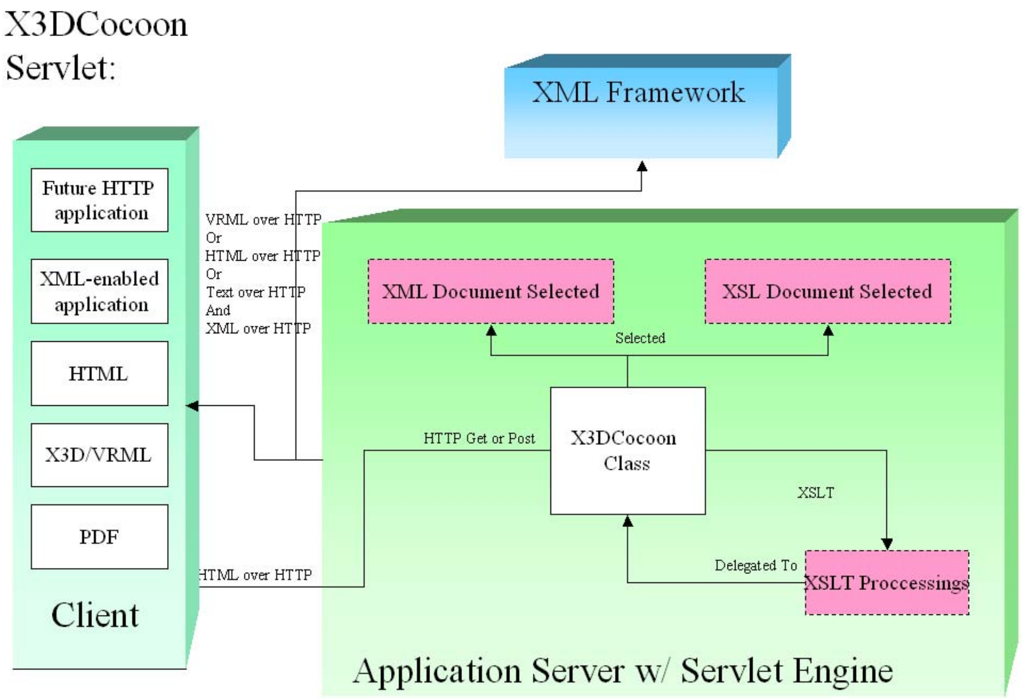


Figure V.4 X3D Cocoon Servlet Architecture

D. X3DCOCOON SERVLET USAGE IN THE WEB PUBLISHING FRAMEWORK.

The X3DCocoon architecture description starts with a selection XHTML page. The X3DCocoon Selection Servlet XHTML Input Form provides an HTML over HTTP interface to an X3DCocoon Servlet (see Appendix G, Section B for the associated source code). The Servlet implants the javax.Servlet.http.httpServlet.class. This class also uses the doGet() and doPost() methods. The doPost() method overrides the doGet() method by simplifying the call to doGet(). The X3DCocoon Servlet get() the X3D XML document along with XSL stylesheet selected from the input form.

The Servlet then calls the DOM method using "XSLTProcessor proc = XSLTProcessorFactory.getProcessor();" and the "proc.process(**new** XSLTInputSource(**new** java.io.StringReader(output)), **new** XSLTInputSource(xsltSystemId), **new** XSLTResultTarget(fileout));" functions. The DOM parses the "output" XML document with the "xsltSystemID" XSL Stylesheet. The method returns the desired serialized results "fileout" to respond to the HTTP request.

The X3Dcoocon Servlet received HTTP request and selects the correct XML documents. The X3Dcoocon Servlet can apply multiple or chained XSL stylesheets (Figure V.5). The Servlet returns the selected formatted output based on the user selected stylesheet. The XSL stylesheet process may involve intermediate steps converting from X3D XML to VRML then to converts the VRML back to XML to be used in the XML-Based Web Publishing Framework. The X3D to Cocoon Servlet uses the X3Dcoocoon.xsl to control the development of a VRML output. This selection then tells the Servlet to apply a wrapper tagset around the VRML outputs. The X3Dcoocoon Servlet selects the X3dToVrml97.xsl stylesheet. The xsltSystemID is changed to X3dToVrml97.xsl to give the user a selection control. The X3Dcoocoon Servlet store the XML document in the correct XML–based framework location and responses to the client request with a text over HTTP (Figure V.6) output.

A text example using X3DCocoon Servlet, which applies the X3dToVrml97.xsl stylesheet(Appendix G, Section K) to the HelloWorld.XML document are rendered using

Internet Explorer (Figure V.6). The stored XML document is used in the XML-Based Framework to generate VRML from XML and describe in Section D of this Chapter. The X3DCocoon Servlet can also convert a XML Document to HTML Appendix G, Section E by applying the X3dtoHTML.xsl stylesheet (Appendix G, Section D). The HTML example using X3DCocoon Servlet, which applies the X3dtoHTML.xsl stylesheet to the HelloWorld.XML document are rendered using Internet Explorer (Appendix G Section F).

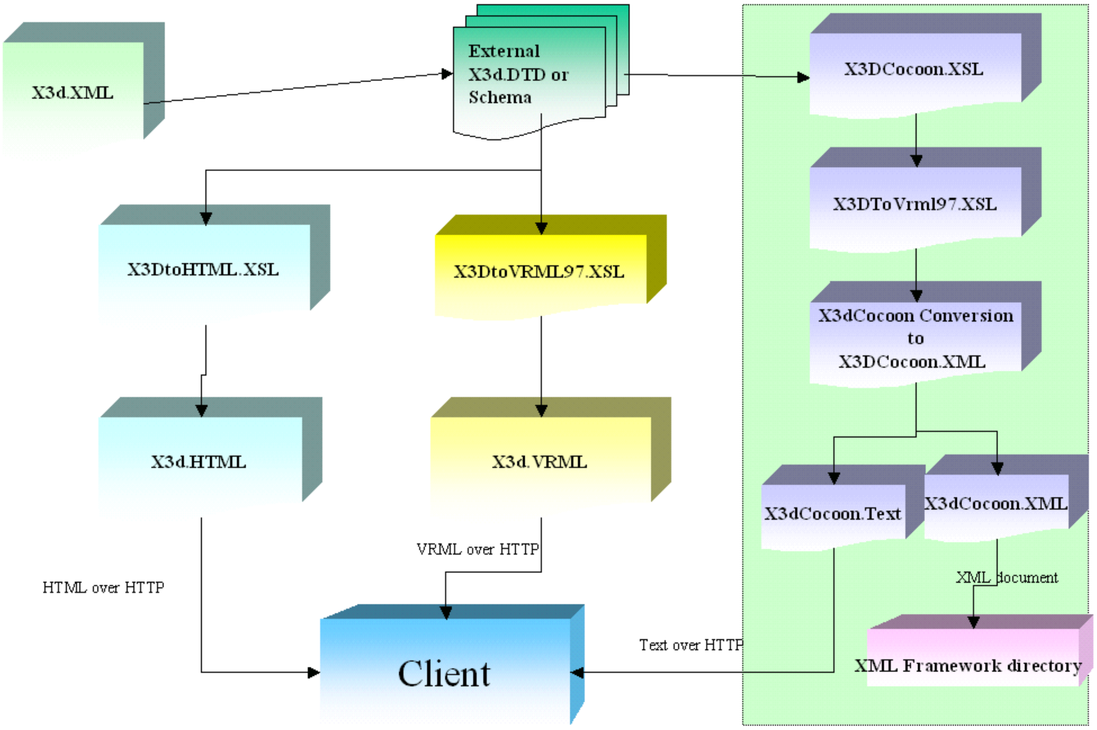
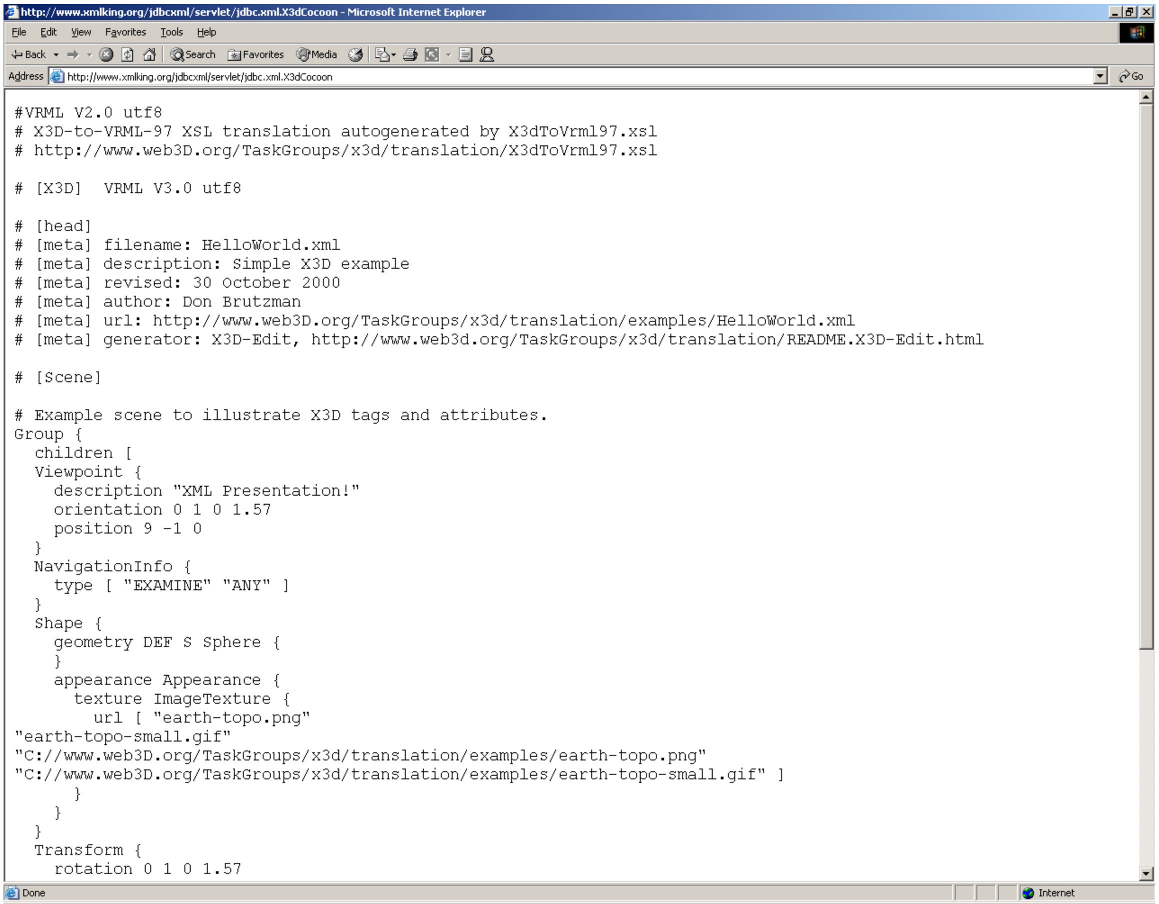


Figure V.5 X3DCocoon Servlet Multiple XML document Processing

The Servlet can convert a XML Document to VRML (Figure V.8) by applying the X3dtoVRML97.xsl stylesheets (Appendix G, Section J). The VRML Example using X3DCocoon Servlet, which applies the X3dtoVRML97.xsl stylesheet to the HelloWorld.XML document (Appendix G Section D) are rendered using Cosmo Player in figure V.8.



http://www.xmlking.org/jdbcxml/servlet/jdbc.xml.X3dCocoon - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address http://www.xmlking.org/jdbcxml/servlet/jdbc.xml.X3dCocoon
Go
Done
Internet

```
#VRML V2.0 utf8
# X3D-to-VRML-97 XSL translation autogenerated by X3dToVrml97.xsl
# http://www.web3D.org/TaskGroups/x3d/translation/X3dToVrml97.xsl

# [X3D]  VRML V3.0 utf8

# [head]
# [meta] filename: HelloWorld.xml
# [meta] description: Simple X3D example
# [meta] revised: 30 October 2000
# [meta] author: Don Brutzman
# [meta] url: http://www.web3D.org/TaskGroups/x3d/translation/examples/HelloWorld.xml
# [meta] generator: X3D-Edit, http://www.web3d.org/TaskGroups/x3d/translation/README.X3D-Edit.html

# [Scene]

# Example scene to illustrate X3D tags and attributes.
Group {
  children [
  Viewpoint {
    description "XML Presentation!"
    orientation 0 1 0 1.57
    position 9 -1 0
  }
  NavigationInfo {
    type [ "EXAMINE" "ANY" ]
  }
  Shape {
    geometry DEF S Sphere {
    }
    appearance Appearance {
      texture ImageTexture {
        url [ "earth-topo.png"
"earth-topo-small.gif"
"C://www.web3D.org/TaskGroups/x3d/translation/examples/earth-topo.png"
"C://www.web3D.org/TaskGroups/x3d/translation/examples/earth-topo-small.gif" ]
      }
    }
  }
  Transform {
    rotation 0 1 0 1.57
```

Figure V.6 HelloWorld Text Display

E. FUNCTIONALITY OF THE FRAMEWORK FOR DISTRIBUTING AND VISUALIZING THREE-DIMENSIONAL SIMULATION UTILIZING X3D AND VRML.

XML-Based Web Publishing Framework allows for VRML 3D simulations using the Architecture in figure V.7. The XML-Based Web Publishing Framework generates 3D simulations using FOP and X3DVRML.XSL (Appendix G Section C) developed for the prototype system. This section documents X3D and VRML three-dimensional simulations using XML-based Web Publishing Framework.

The XML document is validated against an internal and external DTD. The framework automatically responds to client's XML document request by using the XML document extension and document name selected by the user such as "CocHelloWorld.xml".

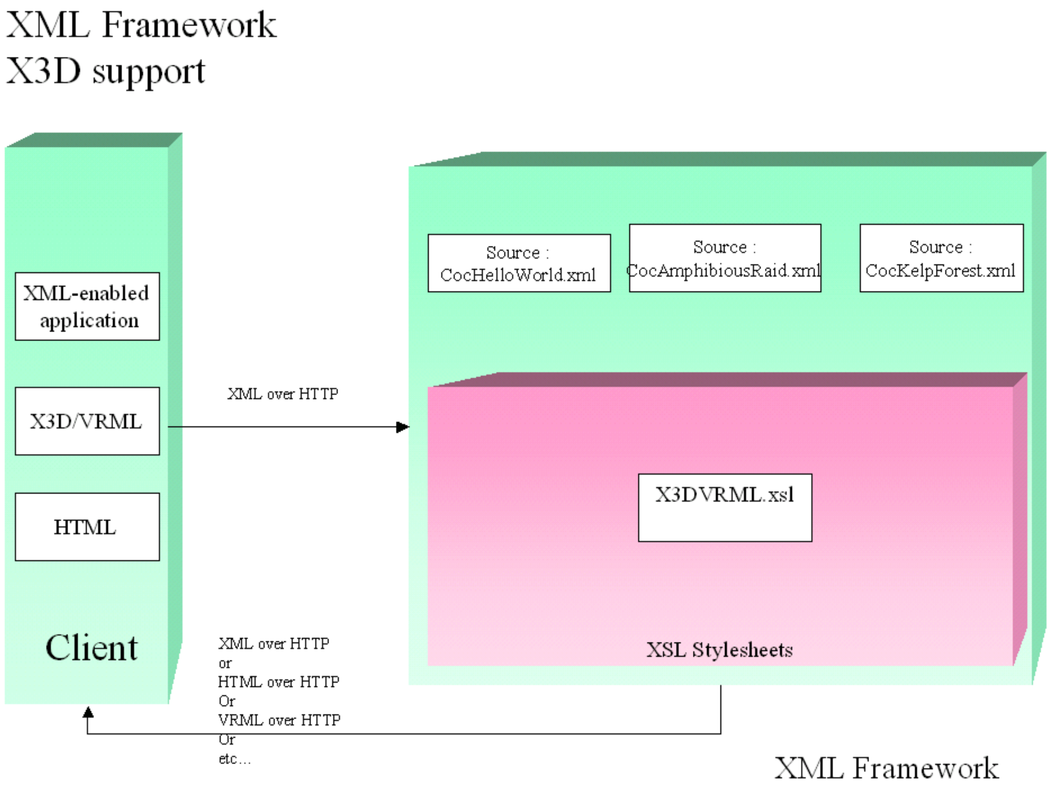
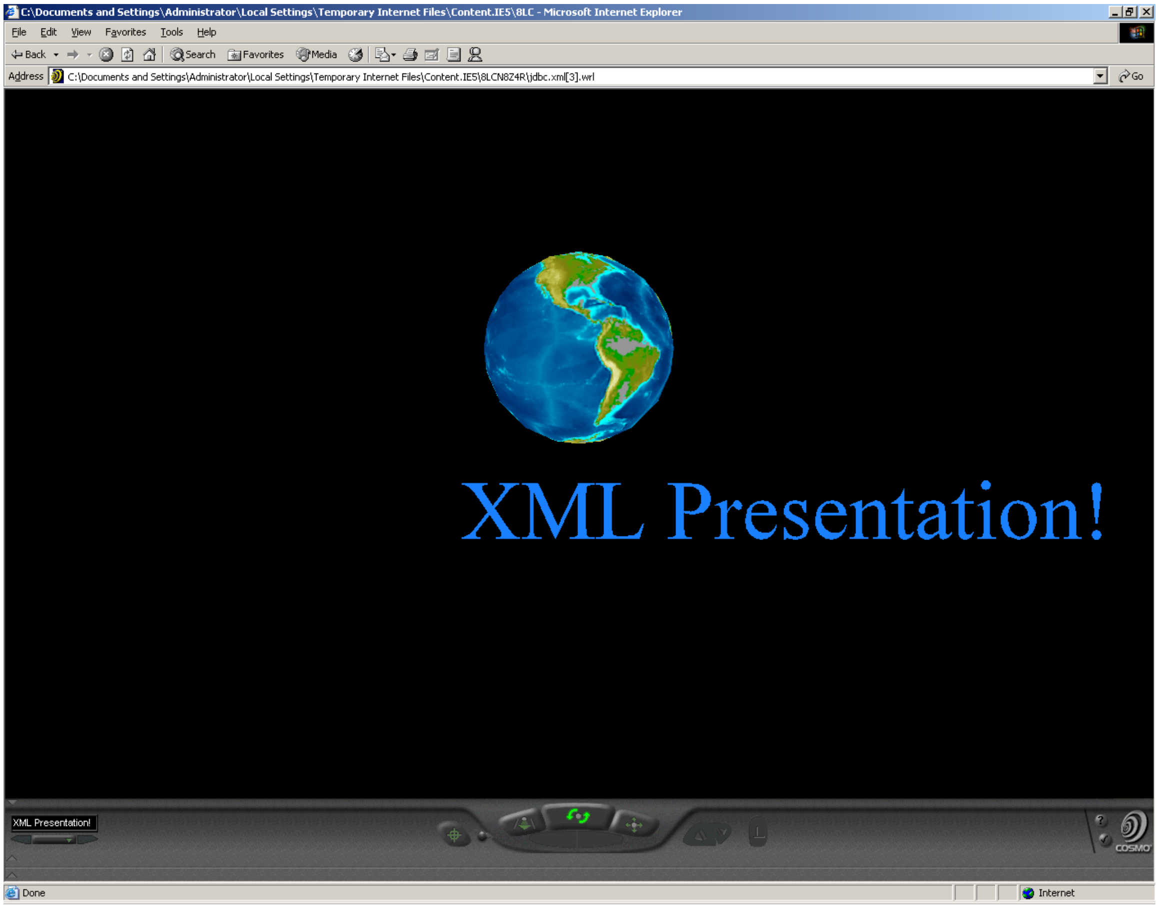


Figure V.7 XML database generic connection architecture

The CocHelloworld.XML (Appendix G Section G) document is requested by the client browser via a call to the XML document using the Web Publishing Framework URL and document name. The framework, in response to the users request, validates XML document against a DTD and calls several processes using XSP and XSP: FO namespaces and converts the XML data using the X3DVRML.XSL stylesheet (Appendix G, Section C). The Framework renders the output in VRML over HTTP to clients Web browser (Figure IV.8).


C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\8LC - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Back
Search
Favorites
Media
Go
XML Presentation!
XML Presentation!
COSMO
Done
Internet

Address C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\8LCN8Z4R\jdbc.xml[3].wrl

Figure V.8 HelloWorld or CocHelloWorld VRML display

The Web Publishing Framework prototype site uses XML and Java technologies in the Web site is design. The CocKelpForest.XML (Appendix G Section H) document is requested by the client browser via a call to the XML document using the Web Publishing Framework URL and document name. In this example the CocKelpForest.XML document was edit to test The XML-Based Web Publishing Framework's, ability to response to the users request, calls several processes using XSP and XSP: FO namespaces and converts the XML data using the X3DVRML.XSL stylesheet (Appendix G, Section C). The XML document internal links were change to call only use stored VRML files to render the full scene. The Framework returns VRML over HTTP to the Clients Web browser (Figure V.9).

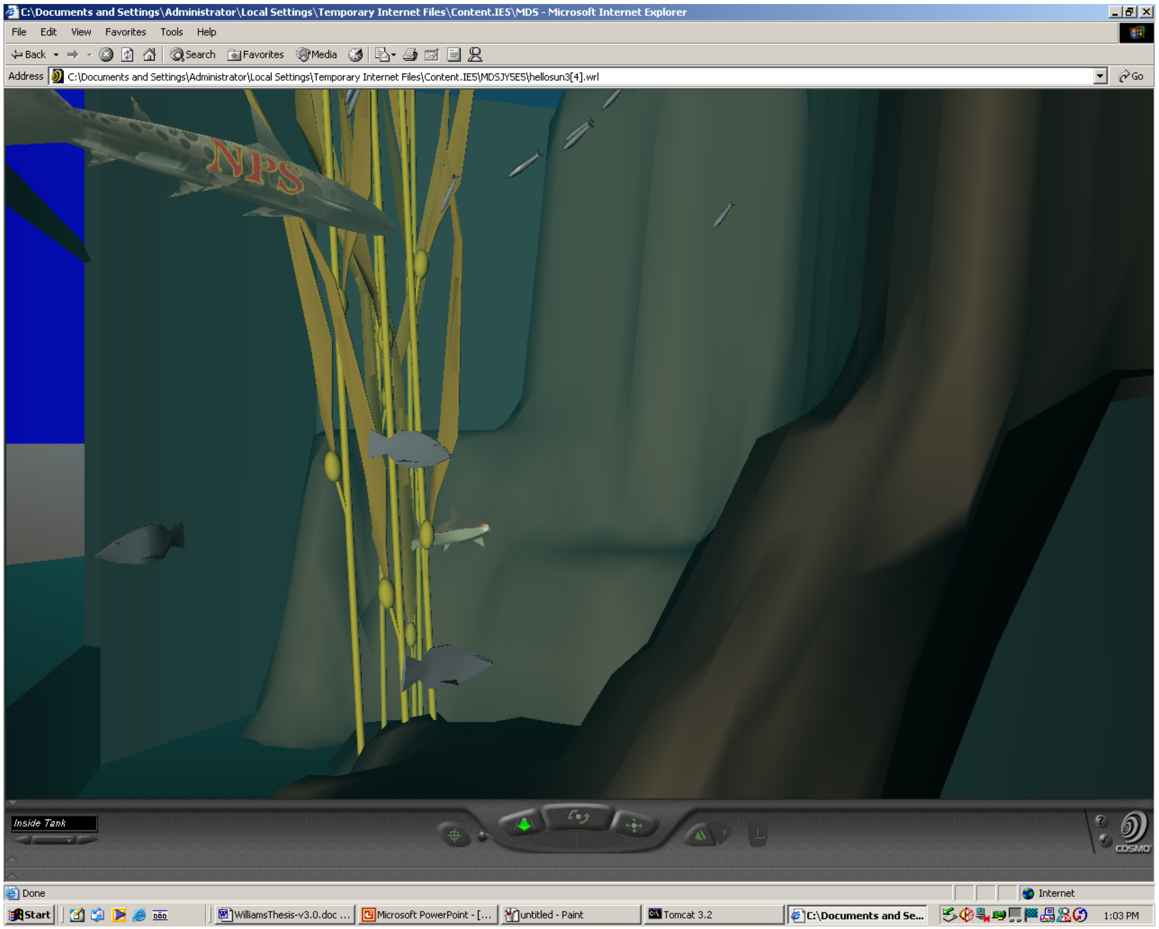
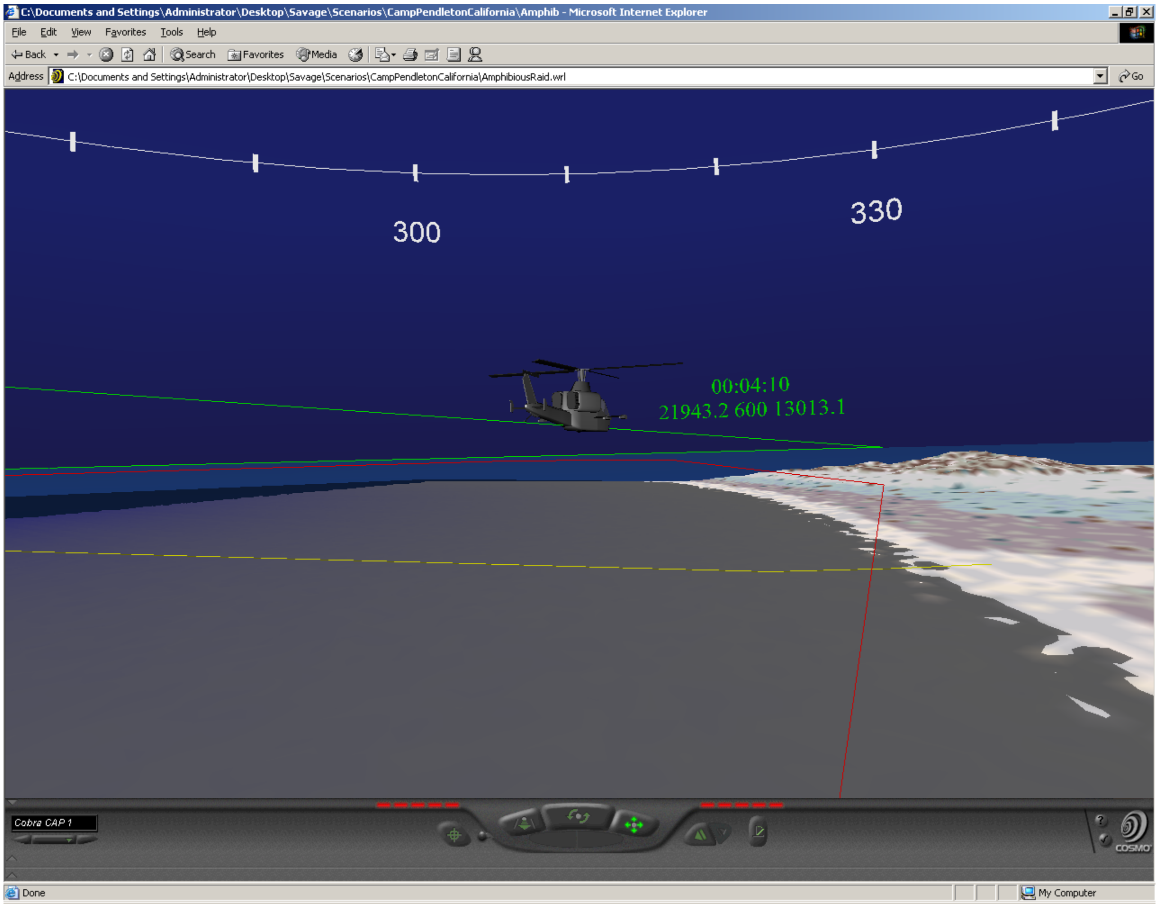

C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\MDS - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\MDSJY5E5\hellosun3[4].wrl
Go
NPS
Inside Tank
COSMO
Done
Internet
Start
WilliamsThesis-v3.0.doc ...
Microsoft PowerPoint - [...
untitled - Paint
Tomcat 3.2
C:\Documents and Se...
1:03 PM

Figure V.9 CocKelpForest VRML display

The Web Publishing Framework prototype site uses XML and Java technologies in the Web site is design. The CocAmphibiousRaid.XML (Appendix G Section I) document is requested by the client browser via a call to the XML document using the Web Publishing Framework URL and document name. In this example, the CocAmphibiousRaid document was not edited. The Based-Based Web Publishing Framework responded to the users request, called several processes using XSP and XSP: FO namespaces and converted the XML data using the X3DVRML.XSL stylesheet (Appendix G, Section C). The Framework returned VRML over HTTP to the client's Web browser (Figure V.10).



C:\Documents and Settings\Administrator\Desktop\Savage\Scenarios\CampPendletonCalifornia\Amphib - Microsoft Internet Explorer

Address C:\Documents and Settings\Administrator\Desktop\Savage\Scenarios\CampPendletonCalifornia\AmphibiousRaid.wrl

Figure V.10 CocAmphibiousRaid VRML display

F. SUMMARY

The XML-based Web Publishing Framework supports 3D simulations utilizing X3D and VRML. X3D overcomes some of the limitations of the original VRML specification and incorporates the use of the Extensible Markup Language (XML). These technologies are scalable within the XML-based Web Publishing Framework architecture. This approach is suitable for the generation and distribution of large-scale virtual environments.

The XML-based Web Publishing Framework, allowed for the separation between application programming and Web presentation. The incorporation of a Model, View, and Controller (MVC) design approach provided a clean separation between different components (logic, presentation, and data) of information and the programming functionality. The XML-based Web Publishing Framework separates the Web developer's from the X3D and/or Java developers'.

The XML-Based Web Publishing Framework supports 3D modeling, addressed the XML-based data exchange and the developed an on fly X3D simulation presentations. The XML-based Web publishing Framework reduces the need to store multiple version of the same output. The sites only need to store a single stylesheet for each output and the XML documents. The framework reduces complexity in site maintenance.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSIONS AND RECOMMENDATIONS

A. INTRODUCTION

This chapter presents conclusions drawn from the Network Application Server using Extensible Mark-Up Language (XML) to support distributed databases and 3D environments. Recommendations for future research related to this thesis are also presented.

B. CONCLUSIONS

The approach adopted in this thesis used an XML-based Web Publishing framework as a more efficient means of providing systems interoperability, and demonstrated the ability of XSL stylesheets to enable automated translations between XML and desired system outputs. This approach also allows the DoD to lower system integrations costs. The framework demonstrated the ability to view each system separately and deal only with the interfaces. The use of these technologies reduces Web site development and maintenance costs.

This research demonstrated the functionality of an XML-based Web Publishing framework, which allows for the separation between application programming and Web presentation. This incorporation of a Model, View, and Controller (MVC) design approach provides a clean separation between different components of information (logic, presentation, and data) and the programming functionality. The Web framework provides the ability to distribute many different forms of data across heterogeneous systems.

The XML-based Web Publishing framework also provides the ability to support the WEN architecture infrastructure. Web application framework easily supports distributed databases and XML-based presentations, including 3-D simulations utilizing X3D and VRML.

This research addressed the problem of sharing information between two or more systems using the XML-based Web Publishing framework. The prototype demonstrates the framework's capability to extend relational database management systems (RDBMS) to support the XML-based Web technology. The need for an XML-based database interface was motivated by advances in technology, specifically object-oriented architecture, Java programming language, and middleware technologies, and a concerted effort on the part of Navy to use NXI as part of the WEN scope. The Web Publishing framework supports distributed systems that employ some of the latest technological tools. This solution highlights the use of Java, XML and other Java technologies to implement a truly scalable, distributive and cost-effective system. In the development of the Web Publishing framework system, this research evaluated several developmental tools very carefully before selecting the Cocoon platform for prototype testing. The disadvantage of Java is in the slow execution speed of the byte code, but for the WEN application, speed over Internet connection should not be a major concern. The WEN application is based on the TCP/IP WAN model and more importance should be attached to the completeness of a transaction than to speed. The Navy needs systems that are totally heterogeneous and that can reliably do distribution with minimal blocking on method invocations.

The XML-based Web Publishing framework supports three-dimensional modeling, addressing the XML-based data exchange, and then developing "on the fly" X3D simulation presentations. The XML-based Web Publishing framework client response is created during document production. The client's browser performance is dramatically improved since clients start receiving data as soon as the Framework creates it, rather than after all processing stages are completed. The Web publishing Framework reduces the need to store multiple versions of the same output. The sites only need to store a single stylesheet for each output and the XML documents. The current web 3D design [http://www.web3d.org/TaskGroups/x3d/translation/examples/contents.html] stores multiple versions of the same data in different document formats (HTML, VRML, X3D). The framework reduces complexity in site maintenance. A developer no longer needs to track, update and maintain multiple versions of this data in different formats. These functions are accomplished by just changing one X3D.XML document. The XML-based Web Publishing framework automatically handles the document type of format response to the user request. Since most of the server processing required in Cocoon is incremental, an incremental model allows XML production events to be transformed directly into output events with character streaming, thus avoiding the need to store them in memory. Reduced memory needs allow a greater number of concurrent operations to take place simultaneously, thus allowing the publishing system to scale as the load increases [Axml01].

C. RECOMMENDATIONS FOR FUTURE WORK

The XML-based Web Publishing Framework supports an unlimited amount of Web applications services; all of the constraints within this research were imposed to limit the scope of this thesis to demonstrate an effective proof of principle. Future work should focus on the following:

- The incorporation of the Web Publishing Framework in an Enterprise Environment using Enterprise Java Beans. Using EJB in XML Based Web Publishing Framework Enterprise Environment should be a matter of setting the Servlet environment to acts a client EJB. So on one side EJB server running and listening for upcoming request, and on the other side, it starts the Servlet container with the right EJB client libraries so that whatever Servlet (or Cocoon code through XSP page for instance) the user writes can contain EJB invocations.
- The continued development of an XML schema and XSL stylesheet that automatically generates a 3D visualization of the operational scenarios. Work is ongoing and close to completion. Continued work and collaboration with the XML-MTF Development Team, Naval Undersea Warfare Command, and Institute for Defense Analysis will ensure a robust schema is created that combines elements of USMTF and the XML based Web Publishing Framework. Once the scenario can be, auto generated in 3D, it can be extended in functionality and complexity of the server to provide presentation as requested for planning or post operational reviews.
- The development of a distributed database to support X3D data storage. The X3D virtual world should stream line the current data and storage to reduce complexity of the system in the current a data structure. The use of distributed database storages would increase the system ability to support current scenarios and future scenarios that are more complex. As 3D models are created to include additional ships, tanks, bulldozers, aircraft, motor transportation assets, weapons systems, individual weapons, artillery, and realistic humanoids the database would act as a repository for standardized X3D objects.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. WEB PUBLISHING FRAMEWORK EVALUATION

A. INTRODUCTION

As of the date of this research (November 2001), The available publishing frameworks, which support the requirements, are limited. One of the best resources currently available products is XML Software's list at http://xmlsoftware.com/publishing/. An evaluation rating was applied to frameworks resource (Section B).

B. RESOURCE

Products	**Evaluation Rating**
Resin	
Version: 1.2	Good
Developer: Caucho Technology	
License: Open source	Good
Platforms: JSP	Limited
Resin creates Web applications with server side JavaScript and JSP 1.0. Its focus lies on server page generation with XSLT.	Limited
Includes XML DOM parser XPath queries and, XSLT engine	Good
Supports JSP tag libraries with XSLT	Good
Java HTTP/1.1 Web server	Good
JSDK 2.1 Servlet-engine	Good
Supports Unix and Windows Web servers	Good
Supports July XSL draft (?)	??
XML Template Pages creates full-powered JSP pages from simple XML source	Limited
JSP tag libraries are easy to create with XTP	Good
XSLT-lite simplifies the design of XSL stylesheets	Good

X4Dialog	
Version: 1.0	Early, Not Considered
Developer: Oliver Bruening	
License: Free for non-commercial use	Not Considered
Platforms: Sun JDK 1.1.6 or higher	Good
Java-based XML/XSLT publication framework for dynamic Web sites.	
Relevant	
Version: ?	Early, Not Considered
Developer: Ensemble	
License: Commercial	Not Considered
Platforms:	??
XML-powered dynamic Web publishing tool. Creates HTML pages on the fly using templates, can email or fax retrieved information, and supplies an e-commerce solution. Content management features include authorization/access control and purging of out-of-date information. Creates valid or well-formed XML documents.	
TalvaStudio	
Version: Beta 1	Early, Not Considered
Developer: Talva corporation	
License: Free	No Source Code
Platforms: WWW	??
Checkout, edit, check-in, lock, rename,, copy, delete files. Text files are automatically indexed (full text indexing) in real time, as they are included in the	

netfolder. Manage your XML server with administration tools. Access a J2EE development environment to dynamically produce your XML documents. Use a rich library of XML based Web services that you can re-use in your applications.	
Frontier	
Version: 5.1.6	Good
Developer: Userland	
License: Commercial; free evaluation available	Not Considered
Platforms: Macintosh and Windows	Limited, Not Considered
Frontier 5.1 is a cross-platform content management system for Windows and Macintosh. It features a powerful Website framework, high-performance object database, multi-threaded runtime, and system-level scripting system.	
XML Portal Server	
Version: ?	Early, Not Considered
Developer: Sequoia Software Company	
License: Commercial; eval license available	Not Considered
Platforms: Windows NT	Limited, Not Considered
The XML Portal Server provides an information delivery mechanism powered by XML. Offers strong search and retrieval capabilities. You can connect it to any business application and streamline data to and from the portal.	
Hiawatha	
Version: 0.1	Early, Not Considered
Developer: Paul Tchistopolksii	
License: Open source	Good

Platforms: Java	Good
Lightweight Web server "for those who are fluent in XSL". Standalone Web server with support for XML, XSL (T), XSLScript, HTTP, and CGI (GET and POST). The distribution includes Saxon XSLT processor. Product Page	
Target 2000	
Version: 4.1	Good
Developer: The Magnus Group, Inc.	
License: Commercial	Not Considered
Platforms: Server: Unix, Windows NT; Client: Windows 95/98, MAC	Good
Target 2000 is an Oracle based product to assist users to repurpose, repackage, a republish subsets of their content for print, on-line, Web, and multimedia. Supported by an easy to use multi-user GUI toolkit which provides: multi-user editing, automatic creation of new products, storage of multimedia data, CDROM creation, fully tagged SGML, XML, or HTML pages, adding and cross-referencing content, Query Tool, archive and change retention, complete redline reports, auto-loading of SGML parsed word processing files; user defined security by user, role, product, shadow files, cloning, spell checking, check in/checkout capability, and integrated work flow. Target also has Internet functionality, a full featured API, and support COM/Corba objects.	
xmltr: XML Translation Suite for Frontier	
Version: 1.1 (1999-01-24)	Early, Not Considered
Developer: The Design Group	
License: Free	No Source Code
Platforms: Frontier	Limited, Not Considered

Xmltr is a suite for UserLand Frontier. You can use xmltr to translate documents marked up with Extensible Markup Language (XML) into alternate representations such as HTML for publication on the Web via the Frontier Web Site Framework or to a tool-specific markup language for print publication.	
UltraXML	
Version: 2.0	Good
Developer: WebX Systems	
License: Commercial	Not Considered
Platforms: Windows 95/98/NT4.0/2000.	Limited, Not Considered
Native WYSIWYG integrated XML editor solution. UltraXML is a comprehensive XML solution that allows document creation, workflow, and publishing. Needs PowerPublisher. Features:	
XML Professional Publisher	
Version: 7.0	Good
Developer: XyEnterprise	
License: Commercial	Not Considered
Platforms: WindowsNT, UNIX and IBM AIX	Good
High performance content formatting and publishing application that composes pages from XML source data and outputs high quality results in XML, PDF, HTML, and traditional PostScript.	
Schemantix	
Version: 0.3	Early, Not Considered
Developer: Schemantix	
License: Open source	Good

Platforms: Java 2; Web server with Servlet and JSP support	Good
System for developing Web applications using XML Schemas as the core representation of application data structures. Currently only supports SOX; XSDL and DTD support coming soon.	
Eclipse	
Version: 1.1	Early, Not Considered
Developer: Chrystal Software	
License: Commercial	Not Considered
Platforms: Microsoft NT 4.0, Solaris 2.6, 2.7	Good
XML-based solution that adds interactive content and personalization to your existing Web site. Eclipse allows online companies to tailor their information delivery. Uses XML- and Java-based applications to manage and assemble individual pieces of Web-based information to create a content management and delivery solution for the Web, print, CD, or wireless devices.	
Xbuilder	
Version: 2.5	Good
Developer: Cape Clear	
License: Commercial; free trial available	Not Considered
Platforms: Windows 95/98/2000/NT	Limited, Not Considered
XBuilder is a tool that speeds up page downloads on database-driven Web sites. This version supports XML and XSL. When XBuilder compiles an XML site, the software transforms the XML and XSL at compile time resulting in HTML that can be served directly to any browser.	

Presenting XML	
Version: 0.5.8	Early, Not Considered
Developer: Daniel Parker	
License: Open source	Good
Platforms: Java	Good
Presenting XML is a framework for Web publishing, which aims to achieve a complete separation of content from presentation. It supports presentation XML content with XSLT stylesheets, as well as additional processing to other formats like PDF. It supports targeting different kinds of media (e.g. browsers, handhelds) with different flavors of markup (e.g. html, WML.)	
Epic (formerly Adept Editor)	
Version: 4.2	Good
Developer: ArborText	
License: Commercial	Not Considered
Platforms: Unix, Windows	Good
A high-end validating SGML/XML authoring and publishing tool for serious document work. Supported XML-related technologies include XSL, XLink, and DOM. Now includes an embedded XSL-FO engine, and a change tracking facility.	
QARE	
Version: 1.0	Early, Not Considered
Developer: Bill la Forge	
License: Open source (LGPL)	Good
Platforms: Java	Good
XML/Java portal (in the form of a Java Servlet), which provides a simple-to-use	

platform to process XML over HTTP.	
TopLeaf	
Version: 5.2	Good
Developer: Turn-Key Systems	
License: Commercial	Not Considered
Platforms: API: Windows 95, 98, NT4, Linux, SCO Unix; GUI: Windows 95, 98, NT4	Good
"TopLeaf is an industrial strength application package for high speed high quality page formatting in the production shop environment. Input data is expected to be marked with XML tagging, under control of the publication's DTD.	
Cocoon	
Version: 2.0 release candidate 1	Good
Developer: Apache Software Foundation	
License: Open source	Good
Platforms: Java	Good
DOM 2.0	Good
Multiple Parser	Good
Supports July XSL draft	Good
SAX 2.0	Good
JAXP 1.1	Good
Publishing framework HTTPServlet that supports the use of DOM, XML, and XSL to provide Web content. The content is created in XML conform a DTD. Cocoon can then process the XML and apply an XSL stylesheet to it for rendering (different stylesheets for different browsers can be used. You can use Cocoon, for example, to create well-formed HTML pages from an XML source. Includes a SAX API and XSLT stylesheet compilation.	

XML ServerWare	
Version: 1.0	Early, Not Considered
Developer: Swift Incorporated	
License: Commercial; free evaluation available	Not Considered
Platforms: Windows + IIS	Limited, Not Considered
Set of Web server components for transformation of XML content on request and for retrieval of XML-formatted data from RDBMS databases. The transformation component works with MSXML 2 and the Microsoft XSL dialect.	
Anastasia	
Version: 1	Early, Not Considered
Developer: Scholarly Digital Editions	
License: Commercial; evaluation version available	Not Considered
Platforms: Windows, Macintosh	Limited, Not Considered
XML publishing software for handling large and highly complex XML documents, where extremely precise control is required over their presentation. It can create output in any format, and it is optimized for HTML output direct to Web browsers. Anastasia permits you to publish documents in identical form both on CD-ROM (Macintosh and Windows) and on the internet, from identical scripts. It includes full support for valid XML and SGML documents, and a fully XML-aware search engine.	
Dynatext	
Version: 4.2	Good
Developer: Enigma, Inc.	
License: Commercial	Not Considered

Platforms: Unix, Windows	Good
Professional publishing system with support for native SGML and XML. You can use it to publish from a single source via any electronic media, such as SGML-based electronic books, CD-ROMs, or HTML or XML pages. Includes publisher-controlled stylesheets, stand-alone browser for viewing, multiple language support, and multiple content views. Supports authoring with Word, Framemaker, and Interleaf.	
IXIASOFT XML Content Server (TEXTML Server)	
Version: 2.0	Good
Developer: IXIASOFT	
License: Commercial; free evaluation version for up to 1000 documents	Not Considered
Platforms: Server: Windows NT 4.0; Client: Internet Explorer 5.0	Limited, Not Considered
The TEXTML Server is an XML document server whose purpose is to store, index, and retrieve large quantities of XML documents. Geared for anyone looking to implement a Web-based XML document management component within a larger solution the TEXTML Server features a documented COM API with an Active Server Page toolkit.	
Dynabase	
Version: 3.1	Good
Developer: Inso	
License: Commercial	Not Considered
Platforms: Microsoft Internet Information Server on Windows NT 4.0, Netscape Enterprise Servers on Windows NT 4.0, Sun Solaris 2.6, Macintosh	Good

Professional application with both powerful content management features and dynamic server environment, aimed at publishing intra/internet Webs. It is designed to handle types of content like HTML, graphics, scripts, applets, and multimedia files, and to work with a variety of editors. Includes an object-oriented repository for version-controlled storage of Web content. Dynabase includes a Web server plug-in and a framework for publishing Web pages on the fly.	
Rocket	
Version: 0.2d	Early, Not Considered
Developer: Micheal Floyd	
License: Free for non-commercial use	Not Considered
Platforms: Windows	Limited, Not Considered
Framework for creating XML-based Web sites. "In a nutshell, Rocket is a collection of skeleton XML documents, XSL style sheets, and DTD's that you can use as a basis for creating your own XML-based Web site. Using Rocket, you can transform XML documents and serve them to any browser, regardless of its capabilities. Rocket also allows you exchange XML streams between XML-capable browsers and HTTP servers. Currently, Rocket is set up to operate with Active Server Pages (ASPs). However, there is nothing to prevent you from dropping the framework into a Servlet environment, or running it in conjunction with Perl's XML: parser module. The one caveat is that some style sheets may have to be tweaked to work with newer style-sheet processors." – From product page.	
Xpublish	
Version: 2.0a-2	Good
Developer: Media Design in Progress	
License: Commercial; free evaluation	Not Considered
Platforms: Macintosh	Limited, Not

	Considered
"XPublish (TM) is a professional Macintosh XML publishing system for efficient development and maintenance of Websites. It works by automatically generate standard HTML pages from XML documents based on a style sheet and processing rules."	
Avenue.quark	
Version: Pre-release	Early, Not Considered
Developer: Quark	
License: Free	No Source Code
Platforms: Mac OS, Windows	Good
Tool that lets you extract the content of QuarkXPress documents and store that content in an XML format specified by a DTD. Tagging rules in Avenue.quark let you associate styles and structures in QuarkXPress documents with XML elements.	
PerXML	
Version: 1.0	Early, Not Considered
Developer: PerCurrence, Inc.	
License: Commercial; free evaluation version available.	Not Considered
Platforms: Windows NT/ME/2000	Limited, Not Considered
XML data transformation and development platform. Includes a runtime engine for transforming native XML and legacy data into XML documents using XSLT transformations. Supports major editing environments such as XML Spy and XMetal Pro. The PerXML Server supports dynamic content integration and delivery in XML, HTML, WML, and most other Web environments.	

APPENDIX B. COCOON INSTALLATION GUIDE

A. INTRODUCTION

Cocoon is a 100% pure Java publishing framework that relies on new W3C technologies (such as XML and XSL) to provide Web content. The Cocoon project aim is to change the way Web information is created, rendered, and delivered. This new paradigm is based on fact that document content, style and logic are often created by different individuals or working groups.

Cocoon's goal is to separate site development into three layers. The three layers allow independent designing, creating and managing, reducing management overhead, increasing work reuse and reducing time to publish.

B. INSTALLATION OF COCOON WEB PUBLISHING FRAMEWORK

The installation of Cocoon is not simple task. This procedure documents how to install the complete Cocoon system. The Cocoon system was loaded on a Windows 2000 Professional operating systems and Windows 2000 Advanced Server operating systems, using Apache Web Server version 1.3.20, Servlet API version 3.2.3, Tomcat Servlet Engine version 3.2.3, the Java Virtual Machine (JVM) SUN JDK version 1.3.1_01 and Cocoon version 1.8.2. The Cocoon Web-publishing framework is an open source system designed to support heterogeneous environments. This Cocoon Web-publishing framework procedure was written with these systems. Cocoon maintains as in Table B.1 listing of configuration which the system were successfully loaded at http://xml.apache.org/cocoon/

Operating System	Web Server	Servlet Engine	JVM
Debian Potato or Woody	Apache 1.3.12	JServ 1.1.2	IBM JDK 1.1.8 or 1.3
Debian Potato	Apache 1.3.14	Tomcat 3.2.1	Sun JDK 1.3
Debian Woody	JWS 2.0	IAS 4.1.1	IBM JDK 1.3

RedHat Linux 6.0	Apache 1.3.9	Apache JServ 1.0	IBM JDK 1.1.8
RedHat Linux 6.0	Apache 1.3.9	Apache JServ 1.1b3	IBM JDK 1.1.8
RedHat Linux 6.0	Apache 1.3.9	Apache JServ 1.0	Blackdown JDK 1.2pre2
RedHat Linux 6.1	Apache 1.3.9	JRun 2.3.3	IBM JRE 1.1.8
RedHat Linux 6.1 (i686)	Apache 1.3.11	Apache JServ 1.1	Sun JDK 1.2.2
RedHat Linux 6.1 (i686)	Apache 1.3.11 + mod_ssl 2.5.0	Apache JServ 1.1	IBM JDK 1.1.8
RedHat Linux 6.1	Apache 1.3.12-2	Jserv 1.1.2-2	JDK 1.2.2_006
RedHat Linux 6.1	Apache 1.3.14	Tomcat 3.2.1	JDK 1.1.6
RedHat Linux 6.2	Apache 1.3.12	JRun 3.0	Sun JDK 1.2.2
RedHat Linux 6.2 (i686)	Apache 1.3.12 + mod_ssl 2.6.5	JRun 2.3.3	Sun JDK 1.2.2
RedHat Linux 6.2	Apache 1.3.12-2	Tomcat 3.1	JDK 1.2.2_006
RedHat Linux 6.2	JWS 2.0	IAS 4.1.1	IBM JDK 1.3
RedHat Linux 6.2 (i686)	Apache 1.3.12	Resin 1.2.b1	Sun JDK 1.2.2
RedHat Linux 7.0	Apache 1.3.12	JServ 1.1.2	Sun JDK 1.3
Linux Mandrake 7.0	Orion/1.0.3		Sun JDK 1.3.0rc1
SuSE 6.2 Linux	Apache 1.3.14 mod_ssl	JServ 1.1.2	Sun JDK 1.1.7 / IBM JDK 1.3
SuSE 6.3 Linux	Apache 1.3.9	Apache JServ 1.1	Sun JDK 1.2.2

SuSE 7.0 Linux (2.2.16)	Apache 1.3.12	Apache JServ 1.1.2, Tomcat 3.1	IBM JDK 1.3
Windows 98	Apache 1.3.9	Apache JServ 1.0	Sun JDK 1.2.2
Windows 98	Apache 1.3.9	Apache JServ 1.0	IBM JDK 1.1.7
Windows 98	Apache 1.3.9	Apache JServ 1.1b3	Sun JDK 1.2.2
Windows 98	Apache 1.3.9	Apache JServ 1.1b3	IBM JDK 1.1.7
Windows 98	MS Personal Web Server	ServletExec 2.2	Sun JDK 1.2.1
Windows 98 SE	Apache 1.3.12	Tomcat 3.1	JDK 1.2
Windows 98 SE	Tomcat 3.2.1		JDK 1.3
Windows NT 4.0	IIS 4.0	ServletExec 2.2	Sun JDK 1.2.1
Windows NT 4.0	IIS 4.0	JRun 2.3.3	Sun JDK 1.2.1
Windows NT 4.0	Apache 1.3.9	JRun 2.3.3	Sun JDK 1.2.2
Windows NT 4.0	Tomcat 3.1 Milestone 1		Sun JDK 1.2.2
Windows NT 4.0 SP3	Apache 1.3.12	JServ 1.1.2	Sun JRE 1.2.2
Windows NT 4.0 SP4	Apache 1.3.12	Tomcat 3.1	Sun JDK 1.3
Windows NT 4.0 SP4	BEA Web Logic Server 5.1 SP3		Sun JDK 1.2.2
Windows NT 4.0 SP5	WLS 6.0 Beta		JDK 1.3
Windows NT 4.0 SP5	IIS 4.0	JRun 3.0	Sun JDK 1.2.2
Windows NT 4.0 SP5	IIS 4.0	Web sphere 3.5 Enterprise	IBM JDK 1.2.2
Windows NT 4.0 SP5	Apache 1.3.14	Tomcat 3.2b8	Sun JDK 1.2.2

Windows NT 4.0 SP6	Apache 1.3.6	JDK 1.1.8	Resin 1.1
Windows NT 4.0 SP6a	Apache 1.3.11	Apache JServ 1.1	Sun JDK 1.2.2
Windows NT 4.0 SP6a	Apache 1.3.12	Tomcat 3.1	Sun JDK 1.2.2
Windows NT 4.0 SP6a	IIS 4.0	Tomcat 3.2.1	Sun JDK 1.3.0
Windows 2000 Professional	Apache 1.3.12	Apache JServ 1.1	Sun JDK 1.2.2
Windows 2000 Professional	Apache 1.3.12	Tomcat 3.1	Sun JDK 1.3
Windows 2000 Professional	Resin 1.1		Sun JERK 1.2.2
Windows 2000 Professional	JWS 2.0	IAS 4.1.1	Sun JDK 1.3
Windows 2000 Professional	Dynamo 4.5.1		Sun JDK 1.2.2
Windows 2000 Advanced Server	Tomcat 3.1		JDK 1.3
MacOS 8.5+	Resin 1.1b		MRJ 2.2
MacOS 8.6	Tomcat 3.1		MRJ 2.2.2
MacOS 8.6	WebSTAR 4.0	JRun 2.3	Mrj 2.1.4
MacOS 8.6	WebSTAR 4.0	ServletExec 2.1	Mrj 2.1.4
MacOS 8.6	Quid Pro Quo 2.1.3	ServletExec 2.2	Mrj 2.1.4
Solaris 2.5.1	Apache 1.3.12	Tomcat 3.1	Sun JDK 1.2.2
Solaris 2.5.1	Netscape-Enterprise/3.6 SP3	ServletExec 2.2	Sun JDK 1.2.1
SunOS Netria 5.6	Apache 1.3.9	Apache JServ 1.1b3	Sun JDK 1.1.7
Solaris 2.6	Apache 1.3.14	JServ 1.1.2	Sun JDK 1.3

SPARC			
Solaris 7	Apache 1.3.14	Locomotive 4.2.0	JDK 1.3
Solaris 7	Netscape Enterprise Server 3.2	JRun 2.3	Sun JDK 1.2
Solaris 8 (SPARC)	Apache 1.3.12	Tomcat 3.1	Sun JDK 1.3 Beta Refresh
Solaris 8 (x86)	Resin 1.2.b1		JDK 1.3 beta
FreeBSD 3.4	Apache 1.3.9	Apache JServ 1.0	Blackdown JDK 1.1.8
FreeBSD 3.4-STABLE with Linux base-6.1 for Linux-emulation	Jetty Java HTTP Server v2.3.3		Blackdown jdk-1.2.2-RC4-linux-i386-glibc
FreeBSD 4.1 (RELEASE)	Apache 1.3.12	Jserv 1.1.2	JDK 1.1.8
FreeBSD 4.1 (RELEASE)	Apache 1.3.12	Jserv 1.1.2	Linux JDK 1.2.2
SCO OpenServer 5	Web Logic 4.5.1		SCO JDK 1.1.7A
OS/2 Warp 4 FP 12	Gefion Software LiteWebServer		IBM JDK 1.1.8
SGI IRIX 6.5.7	Apache 1.3.12	Apache JServ 1.1	Sun JDK 1.2.1 (SGI)

Table B.1 Listing Of Configuration System Using Cocoon

C. INSTALLING JAVA VIRTUAL MACHINE

The first program, which needs download and run is the Java Virtual Machine (JVM) (the JAVA interpreter). The JVM reads byte code and translates the code into a language that the computer can understand, possibly storing data values as the program executes. SUN Microsystems provides the JVM in the Java 2 Software Development Kit package J2SDK-1_3_1_01-win.exe which may be obtained from SUN Microsystems at http://java.sun.com/j2se/1.3/install-windows.html. The Java 2 Platform is self-extracting and load automatically.

D. INSTALLING JAKARTA ANT

The Ant program may or may not be needed to install cocoon. The ANT program may be added depending on the individual system needs and if the framework need to changed and recompiled the system. This procedure was added to assist individuals, which may need to build java source code into binary code for cocoon development. Ant operates similarly to a make file program, but Ant is different. Instead of a model where it is extended with shell-based commands, Ant is extended using Java classes. Instead of writing shell commands, the configuration files are XML-based, calling out a target tree where various tasks are executed. Each task is run as an object that implements a particular Task interface.

The ANT build provide for the ability to be cross-platform - to work anywhere and everywhere. Ant has an <exec> task that allows different commands to be executed based on the OS it is executing on.

The source and binary code are distribution to build the Ant. Ant can be obtained at http://jakarta.apache.org/ant. Jakarta-ant-1.4beta2 needs to downloaded and expanded.

Ant supports a number of optional tasks. An optional task is a task, which typically requires an external library to function. The optional tasks are packaged separately from the core Ant tasks. This package is available in the same download directory as the core Ant distribution. Jakarta-Ant-1.4Beta2-optional.jar may be downloaded and placed in the lib directory of the Ant installation.

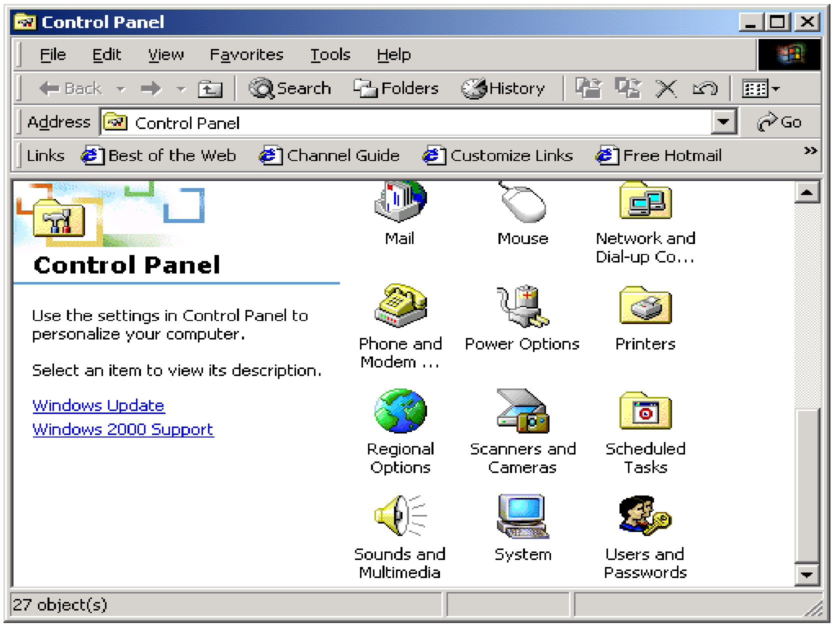
The external libraries required by each of the optional tasks are detailed in the Library Dependencies section. These external libraries may either be placed in Ant's lib directory, where they will be picked up automatically, or made available on the system CLASS PATH environment variable.

Once you have Ant and Java 2 Software Development Kit installed. ANT requires some additional set up which will be needed to be added in Windows 2000:

Assume Ant is installed in c:\jakarta-ant-1.4Beta2. The following sets up the environment:

1. Select: Start|Setting|ControPanel

Control Panel Window should OPEN as in figure B.1.



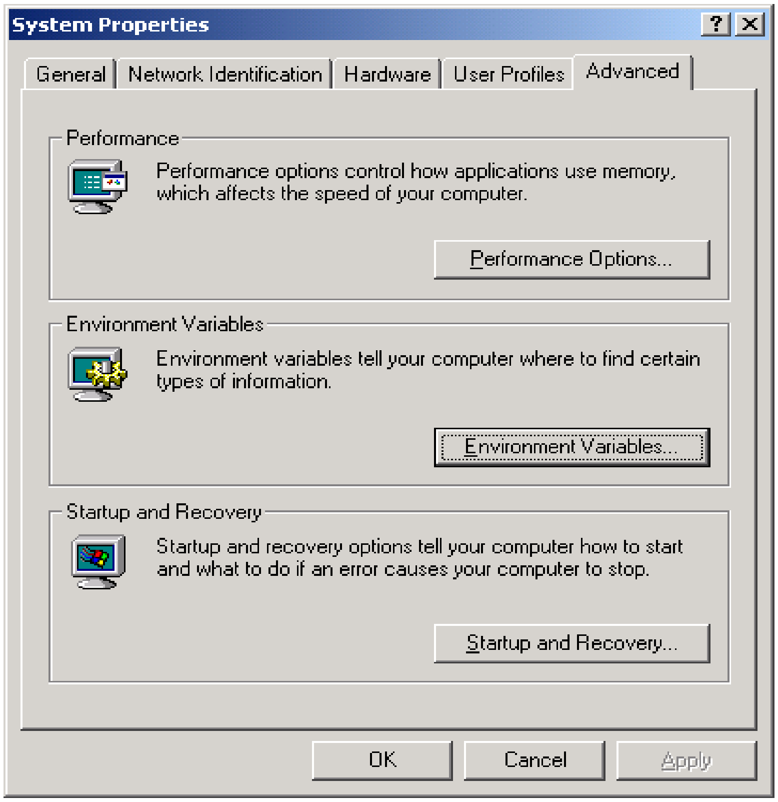
Use the settings in Control Panel to personalize your computer.

Select an item to view its description.

Figure B.1   Windows Control Panel Display

2. Select: System

System Properties Window should OPEN as in figure B.2.



Performance options control how applications use memory, which affects the speed of your computer.

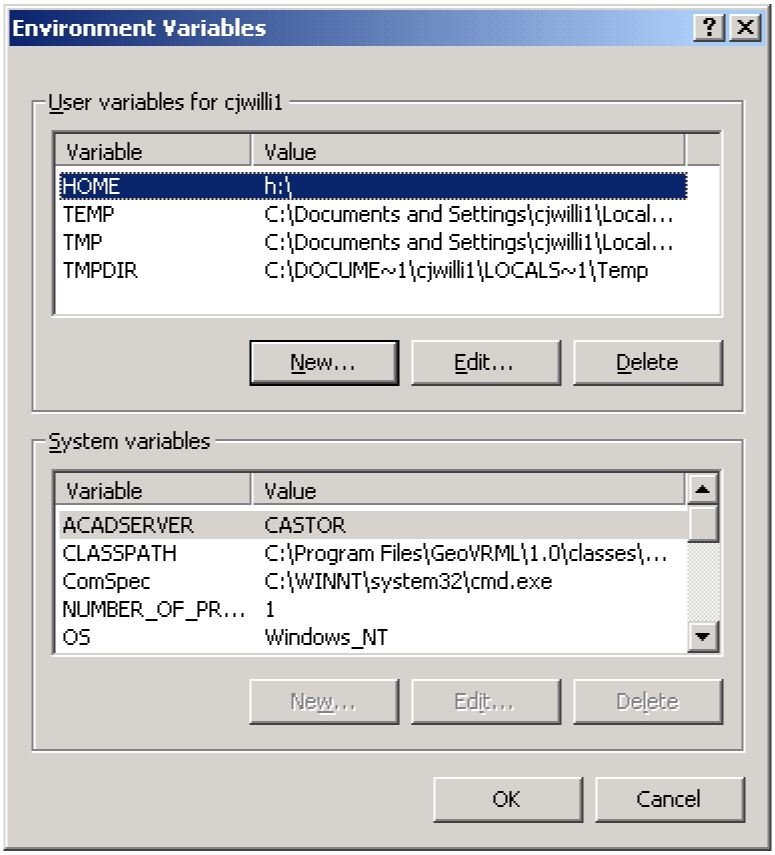
Environment variables tell your computer where to find certain types of information.

Startup and recovery options tell your computer how to start and what to do if an error causes your computer to stop.

Figure B.2 Windows System Properties Display

3. Select: Advance Tab|Enviroment Variables

Environment Variables Window should OPEN as in figure B.3.



Variable	Value
HOME	h:\
TEMP	C:\Documents and Settings\cjwilli1\Local...
TMP	C:\Documents and Settings\cjwilli1\Local...
TMPDIR	C:\DOCUME~1\cjwilli1\LOCALS~1\Temp

Variable	Value
ACADSERVER	CASTOR
CLASSPATH	C:\Program Files\GeoVRML\1.0\classes\...
ComSpec	C:\WINNT\system32\cmd.exe
NUMBER_OF_PR...	1
OS	Windows_NT

Figure B.3 Windows Environment Variables Display

4. If the variables are not listed in the System Variable display window.

4a. Select: New

System Variable display window should open as in figure B.4.

Else

4.b Select: Edit

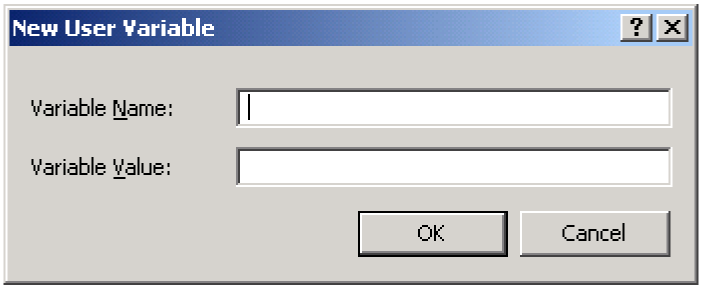


Figure B.4 Windows New User Variables Input Display Panel

Add the bin directory to your path.

5. Variable Name: PATH

6. Variable Value: %PATH%;%ANT_HOME%\bin

7. Select: OK

Add the ANT_HOME environment variable to the directory where you installed Ant.

8. Variable Name: ANT_HOME

9. Variable Value: c:\jakarta-ant-1.4Beta2

10. Select: OK

Add JAVA_HOME environment variable the

11. Variable Name: JAVA_HOME

12. Variable Value: c:\ jdk1.3.1_01

13. Select: OK

14. Select: OK

15. Select: OK

The build.xml file for this ANT distribution has the following targets:

ant build - compiles source files into <ANT_HOME>/lib. This is the default target.

ant jar - creates ant.jar

ant javadoc - creates javadoc documentation in docs/javadoc

ant clean - deletes compiled classes and jar files.

After the environment is setup, the program is ready to build the distribution simply by executing the command:

1. Open a Terminal Window

2. Select:Start|Run

3. Open: cmd

DOS Window should OPEN

4. Change directory to ANT_HOME

5. Type: build

6. Press: Enter

Figure B.5 Show the DOS Window Screen Output

```
C:\jakarta-ant-1.4Beta2>build
... Bootstrapping Ant Distribution
JAVA_HOME=C:\jdk1.3.1_01
JAVA=C:\jdk1.3.1_01\bin\java
JAVAC=C:\jdk1.3.1_01\bin\javac
CLASSPATH=C:\jdk1.3.1_01\lib\tools.jar;lib\crimson.jar;lib\jaxp.jar;
classes;src
\main;
... Compiling Ant Classes
Note: Some input files use or override a deprecated API.
Note: Recompile with -deprecation for details.
... Copying Required Files
src\main\org\apache\tools\ant\taskdefs\defaults.properties
        1 file(s) copied.
src\main\org\apache\tools\ant\types\defaults.properties
```

```
        1 file(s) copied.
... Building Ant Distribution
263 File(s) copied
Buildfile: build.xml
bootstrap:
prepare:
check_for_optional_packages:
build:
Created dir: C:\jakarta-ant-1.4Beta2\build\lib
Compiling 77 source files to C:\jakarta-ant-1.4Beta2\build\classes
Note: Some input files use or override a deprecated API.
Note: Recompile with -deprecation for details.
Copying 1 file to C:\jakarta-ant-1.4Beta2\build\classes
Copying 2 files to C:\jakarta-ant-1.4Beta2\build\classes
Copying 2 files to C:\jakarta-ant-
1.4Beta2\build\classes\org\apache\tools\ant\ta
skdefs\optional\junit\xsl
jars:
Building jar: C:\jakarta-ant-1.4Beta2\build\lib\ant.jar
Building jar: C:\jakarta-ant-1.4Beta2\build\lib\optional.jar
dist-lite:
Created dir: C:\jakarta-ant-1.4Beta2\bootstrap
Created dir: C:\jakarta-ant-1.4Beta2\bootstrap\bin
Created dir: C:\jakarta-ant-1.4Beta2\bootstrap\lib
Copying 2 files to C:\jakarta-ant-1.4Beta2\bootstrap\lib
Copying 8 files to C:\jakarta-ant-1.4Beta2\bootstrap\bin
BUILD SUCCESSFUL
Total time: 9 seconds
... Cleaning Up Build Directories
... Done Bootstrapping Ant Distribution
Buildfile: build.xml
main:
prepare:
check_for_optional_packages:
build:
Copying 2 files to C:\jakarta-ant-1.4Beta2\build\classes
jars:
Building jar: C:\jakarta-ant-1.4Beta2\build\lib\ant.jar
dist-lite:
Created dir: C:\jakarta-ant-1.4Beta2\dist
Created dir: C:\jakarta-ant-1.4Beta2\dist\bin
Created dir: C:\jakarta-ant-1.4Beta2\dist\lib
Copying 2 files to C:\jakarta-ant-1.4Beta2\dist\lib
Copying 8 files to C:\jakarta-ant-1.4Beta2\dist\bin
BUILD SUCCESSFUL
Total time: 3 seconds
```

Figure B.5 ANT Output

E. INSTALLING APACHE

The Apache Web Server needs to be loaded. Apache is an HTTP server, originally designed for Unix systems. Apache was extended to support Microsoft Windows 2000 version[Ahttp01].

Apache Web server is programmed in the C++ program language. Apache is also open source Web server and provides both source code and executable files for listed Operating System. This procedure will use the latest stable executable Apache 1.3.20 Server version. The system must have the Microsoft Installer version 1.10 installed on the PC before it can install the Apache runtime distributions. Windows 2000 is delivered with the Microsoft Installer version 1.10. Instructions on locating the Microsoft Installer, as well as the binary distributions of Apache, are found at http://httpd.apache.org/dist/httpd/binaries/win32/

Note: prior to Apache version 1.3.17. Apache is distributed with an Install Shield 2.0 .exe file. Windows 2000 will not run the Install Shield package.

Apache .msi file is a self-extracting program, which will prompt the user for:

Webmaster name and company name, and whether or not all users should access Apache as a service, or if the system should be installed to run when Webmaster choose the Start Apache shortcut.

Server name, Domain name, and administrative email account.

The directory to install Apache into the default directory (C:\Program Files\Apache Group\Apache although this can be change to any other directory the user wishes). The program will ask for installation type. The "Complete" option installs everything, including the source code if you downloaded the -with_src.msi package. Choose the "Custom" install if you choose not to install the documentation, or the source code from that package.

Select: apache_1.3.20-win32-src-r2.msi and run the installation program.

Note: For standalone testing of Apache. Apache can be setup on LocalHost URL 127.0.0.1 port 80, Server Name: XMLKING. This configuration will allow users to complete Apache load and get the system completely on line prior to Web deployment or for testing purposes. The configuration maybe changed when the system is up and running for deployment. The Apache server can now be called on port 80 by your favorite browser. The browser calls Apache by looping back all server requests to LocalHost, URL 127.0.0.1

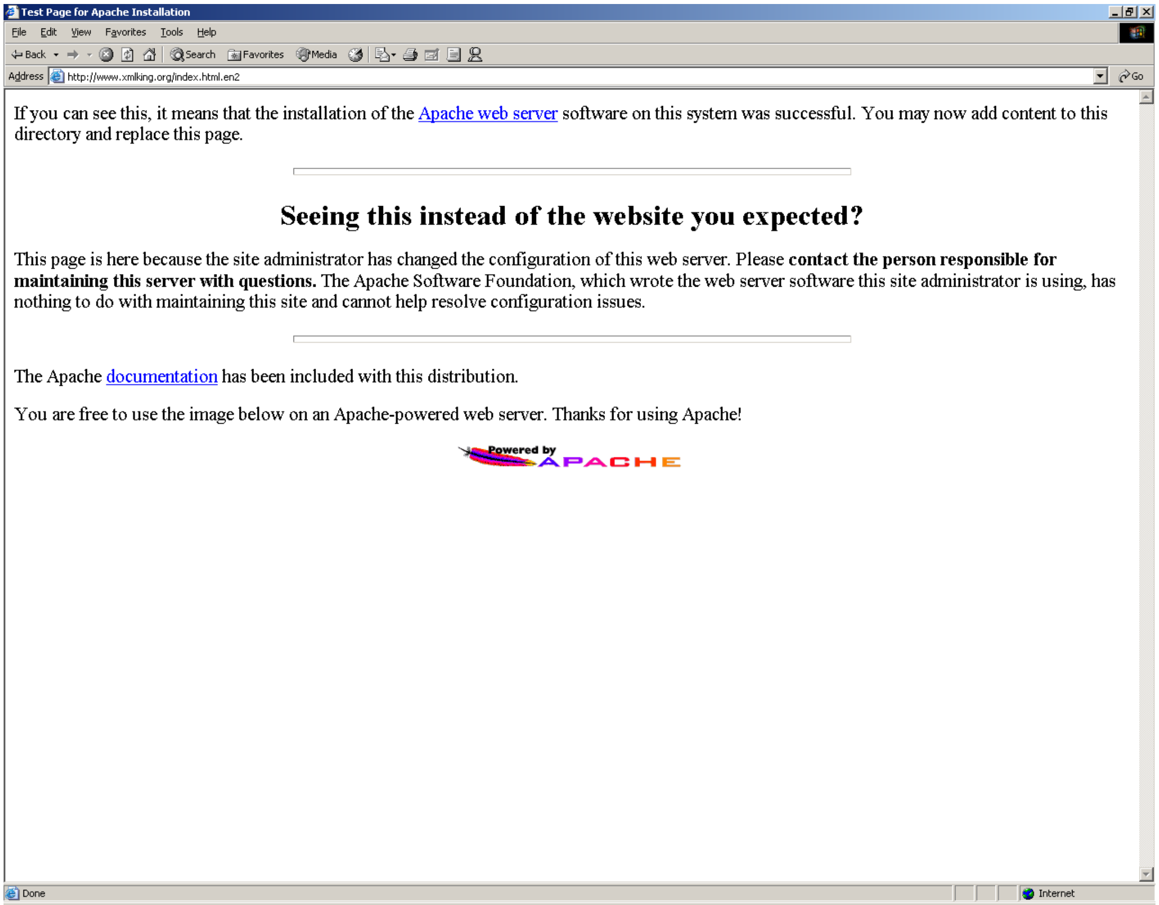
1. Select: Start|Program|Apache Http Server|Control Apache Server|Start

Apache Http Server should startup and display figure B.6 in a DOS Window.

```
Apache service is starting
```

Figure B.6 Apache Output

The user can now test the system by type http://localhost/ on a local browser. Figure B.3 Shows the Initial Apache Startup page.


Test Page for Apache Installation
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
http://www.xmlking.org/index.html.en2
Go
Powered by APACHE
Done
Internet

If you can see this, it means that the installation of the Apache web server software on this system was successful. You may now add content to this directory and replace this page.

Seeing this instead of the website you expected?

This page is here because the site administrator has changed the configuration of this web server. Please **contact the person responsible for maintaining this server with questions.** The Apache Software Foundation, which wrote the web server software this site administrator is using, has nothing to do with maintaining this site and cannot help resolve configuration issues.

The Apache documentation has been included with this distribution.

You are free to use the image below on an Apache-powered web server. Thanks for using Apache!

Figure B.7 Initial startup page

F. INSTALLING JAKARTA-TOMCAT SERVLET ENGINE

Servlet are the analog on the server side to applets on the client side. Servlets are normally executed as part of a Web Sever. Servlets have become a popular and are now supported as part of most major Web servers, including the Netscape Web severs, Microsoft’s Internet Information Server (IIS), the W3C Jigsaw Web server and Apache Web Server.

The source and binary code are distribution for Tomcat Servlet Engine (Jakarta-Tomcat-3.2.3.zip) can be obtained at download directory at http://jakarta.apache.org/downloads. .

Jakarta-Tomcat-3.2.3 requires some additional set up which will be needed for loading Tomcat in Windows 2000[Tom01].:

Assume Tomcat is installed in c:\ Jakarta-Tomcat-3.2.3. The following sets up the environment:

1. Select: Start|Setting|ControPanel

Control Panel Window should OPEN as in figure B.1.

2. Select: System

System Properties Window should OPEN as in figure B.2.

3. Select: Advance Tab|Enviroment Variables

Environment Variables Window should OPEN as in figure B.3.

4. If the variables are not listed in the System Variable display window.

4a. Select: New

System Variable display window should open as in figure B.4.

Else

4.b Select: Edit

Add the bin directory to your path.

5. Variable Name: PATH

6. Variable Value: %PATH%; %TOMCAT_HOME%\bin

7. Select: OK

Setting up a CLASSPATH using system variable display window should open as in figure B.4.

8. If the variables are not listed in the System Variable display window.

8a. Select: New

System Variable display window should open as in figure B.4.

Else

8.b Select: Edit

9. Variable Name: CLASSPATH

10. Variable Value: %TOMCAT_HOME%\classes; (even if does not exist),

11. Select: OK

Setting up a CLASSPATH that contains all of Tomcat dynamically included library jar files. The following jar files are included statically: ant.jar, jasper.jar, jaxp.jar, parser.jar, Servlet.jar, Webserver.jar Setting up a CLASSPATH using system variable display window should open as in figure B.4.

12. Select: Edit

13. Variable Name: CLASSPATH

14. Variable Value: %TOMCAT_HOME%\lib; (even if does not exist),

15. Select: OK

Setting up a CLASSPATH that contains jar file contains the tool javac. Javac contains the files needed for jsp files. Setting up a CLASSPATH, using system variable display window should open as in figure B.4.

16. Select: Edit

17. Variable Name: CLASSPATH

18. Variable Value: %JAVA_HOME%\ lib\tools.jar; (even if does not exist),

19. Select: OK

Setting up a TOMCAT_HOME, environment variable using system variable display window should open as in figure B.4.

20. If the variables are not listed in the System Variable display window.

20a. Select: New

System Variable display window should open as in figure B.4.

Else

20.b Select: Edit

21.Variable Name: TOMCAT_HOME

22.Variable Value: c:\ jakarata-tomcat-3.2.3

23. Select: OK

24. Select: OK

25. Select: OK

Tomcat can now run in a standalone configuration.

The user can start and stop Tomcat using the scripts to start and stop in the bin subdirectory of TOMCAT_HOME. Recommend placing a Shortcut in the Window 2000 Desktop to add better GUI control of Tomcat. Alternative to run is simply by executing the command:

1. Open a Terminal Window

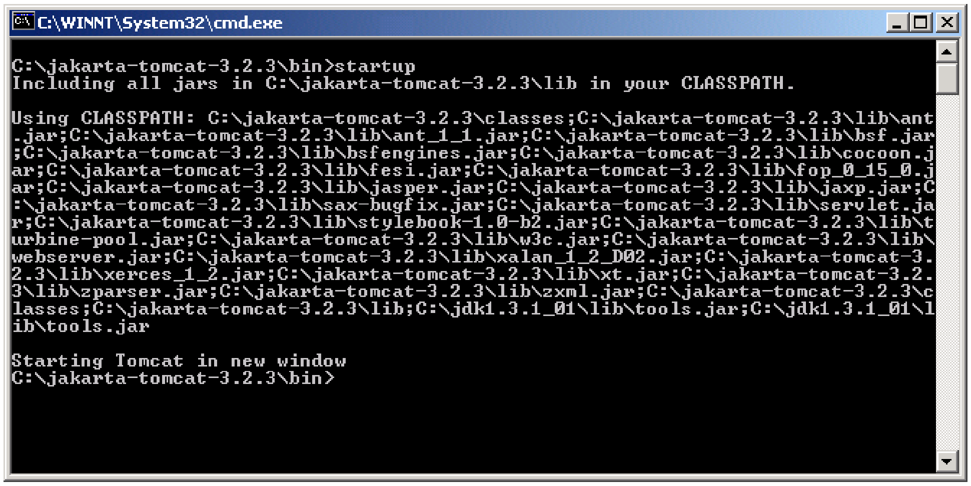
2. Select:Start|Run

3. Open: cmd

DOS Window should OPEN as in figure B.7.

4. To start Tomcat execute:

C:\jakarata-tomcat-3.2.3\bin\startup.bat

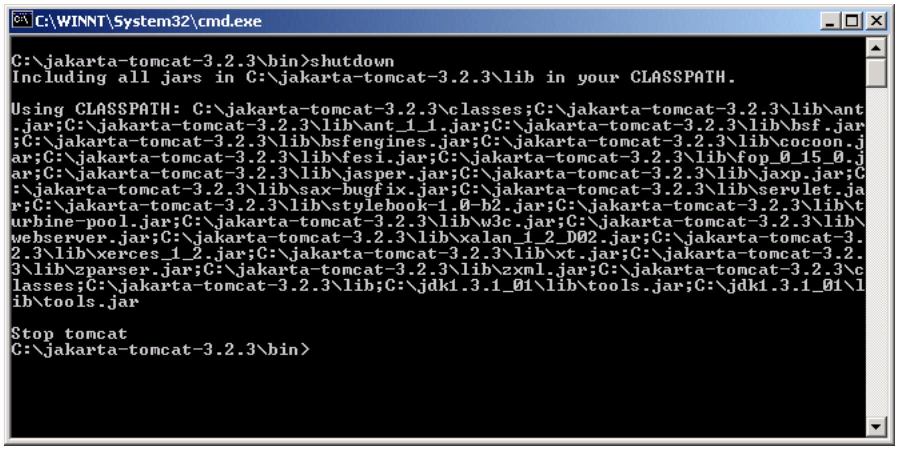


```
C:\jakarta-tomcat-3.2.3\bin>startup
Including all jars in C:\jakarta-tomcat-3.2.3\lib in your CLASSPATH.

Using CLASSPATH: C:\jakarta-tomcat-3.2.3\classes;C:\jakarta-tomcat-3.2.3\lib\ant
.jar;C:\jakarta-tomcat-3.2.3\lib\ant_1_1.jar;C:\jakarta-tomcat-3.2.3\lib\bsf.jar
;C:\jakarta-tomcat-3.2.3\lib\bsfengines.jar;C:\jakarta-tomcat-3.2.3\lib\cocoon.j
ar;C:\jakarta-tomcat-3.2.3\lib\fesi.jar;C:\jakarta-tomcat-3.2.3\lib\fop_0_15_0.j
ar;C:\jakarta-tomcat-3.2.3\lib\jasper.jar;C:\jakarta-tomcat-3.2.3\lib\jaxp.jar;C
:\jakarta-tomcat-3.2.3\lib\sax-bugfix.jar;C:\jakarta-tomcat-3.2.3\lib\servlet.ja
r;C:\jakarta-tomcat-3.2.3\lib\stylebook-1.0-b2.jar;C:\jakarta-tomcat-3.2.3\lib\t
urbine-pool.jar;C:\jakarta-tomcat-3.2.3\lib\w3c.jar;C:\jakarta-tomcat-3.2.3\lib\
webserver.jar;C:\jakarta-tomcat-3.2.3\lib\xalan_1_2_D02.jar;C:\jakarta-tomcat-3.
2.3\lib\xerces_1_2.jar;C:\jakarta-tomcat-3.2.3\lib\xt.jar;C:\jakarta-tomcat-3.2.
3\lib\zparser.jar;C:\jakarta-tomcat-3.2.3\lib\zxml.jar;C:\jakarta-tomcat-3.2.3\c
lasses;C:\jakarta-tomcat-3.2.3\lib;C:\jdk1.3.1_01\lib\tools.jar;C:\jdk1.3.1_01\l
ib\tools.jar

Starting Tomcat in new window
C:\jakarta-tomcat-3.2.3\bin>
```

Figure B.8 DOS Window Tomcat startup.bat

5. To stop Tomcat execute:

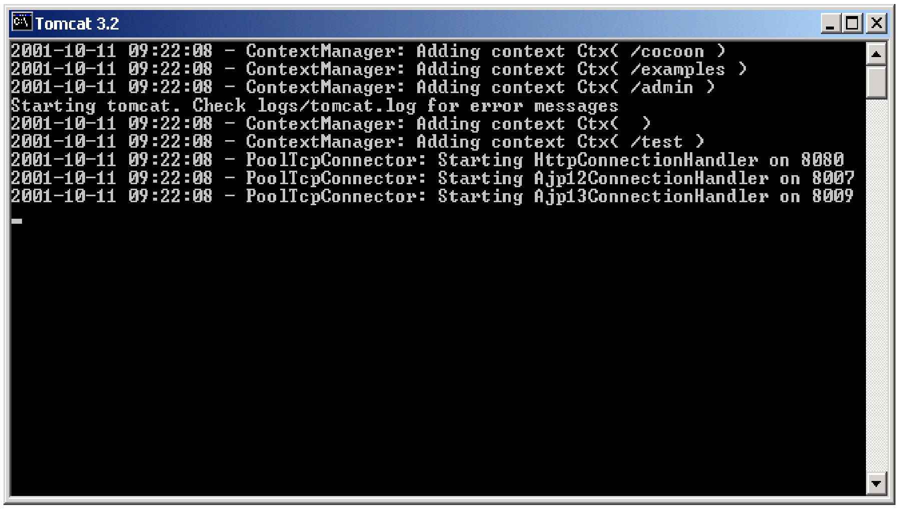
   C:\jakarata-tomcat-3.2.3\bin\shutdown.bat



```
C:\jakarta-tomcat-3.2.3\bin>shutdown
Including all jars in C:\jakarta-tomcat-3.2.3\lib in your CLASSPATH.

Using CLASSPATH: C:\jakarta-tomcat-3.2.3\classes;C:\jakarta-tomcat-3.2.3\lib\ant
.jar;C:\jakarta-tomcat-3.2.3\lib\ant_1_1.jar;C:\jakarta-tomcat-3.2.3\lib\bsf.jar
;C:\jakarta-tomcat-3.2.3\lib\bsfengines.jar;C:\jakarta-tomcat-3.2.3\lib\cocoon.j
ar;C:\jakarta-tomcat-3.2.3\lib\fesi.jar;C:\jakarta-tomcat-3.2.3\lib\fop_0_15_0.j
ar;C:\jakarta-tomcat-3.2.3\lib\jasper.jar;C:\jakarta-tomcat-3.2.3\lib\jaxp.jar;C
:\jakarta-tomcat-3.2.3\lib\sax-bugfix.jar;C:\jakarta-tomcat-3.2.3\lib\servlet.ja
r;C:\jakarta-tomcat-3.2.3\lib\stylebook-1.0-b2.jar;C:\jakarta-tomcat-3.2.3\lib\t
urbine-pool.jar;C:\jakarta-tomcat-3.2.3\lib\w3c.jar;C:\jakarta-tomcat-3.2.3\lib\
webserver.jar;C:\jakarta-tomcat-3.2.3\lib\xalan_1_2_D02.jar;C:\jakarta-tomcat-3.
2.3\lib\xerces_1_2.jar;C:\jakarta-tomcat-3.2.3\lib\xt.jar;C:\jakarta-tomcat-3.2.
3\lib\zparser.jar;C:\jakarta-tomcat-3.2.3\lib\zxml.jar;C:\jakarta-tomcat-3.2.3\c
lasses;C:\jakarta-tomcat-3.2.3\lib;C:\jdk1.3.1_01\lib\tools.jar;C:\jdk1.3.1_01\l
ib\tools.jar

Stop tomcat
C:\jakarta-tomcat-3.2.3\bin>
```

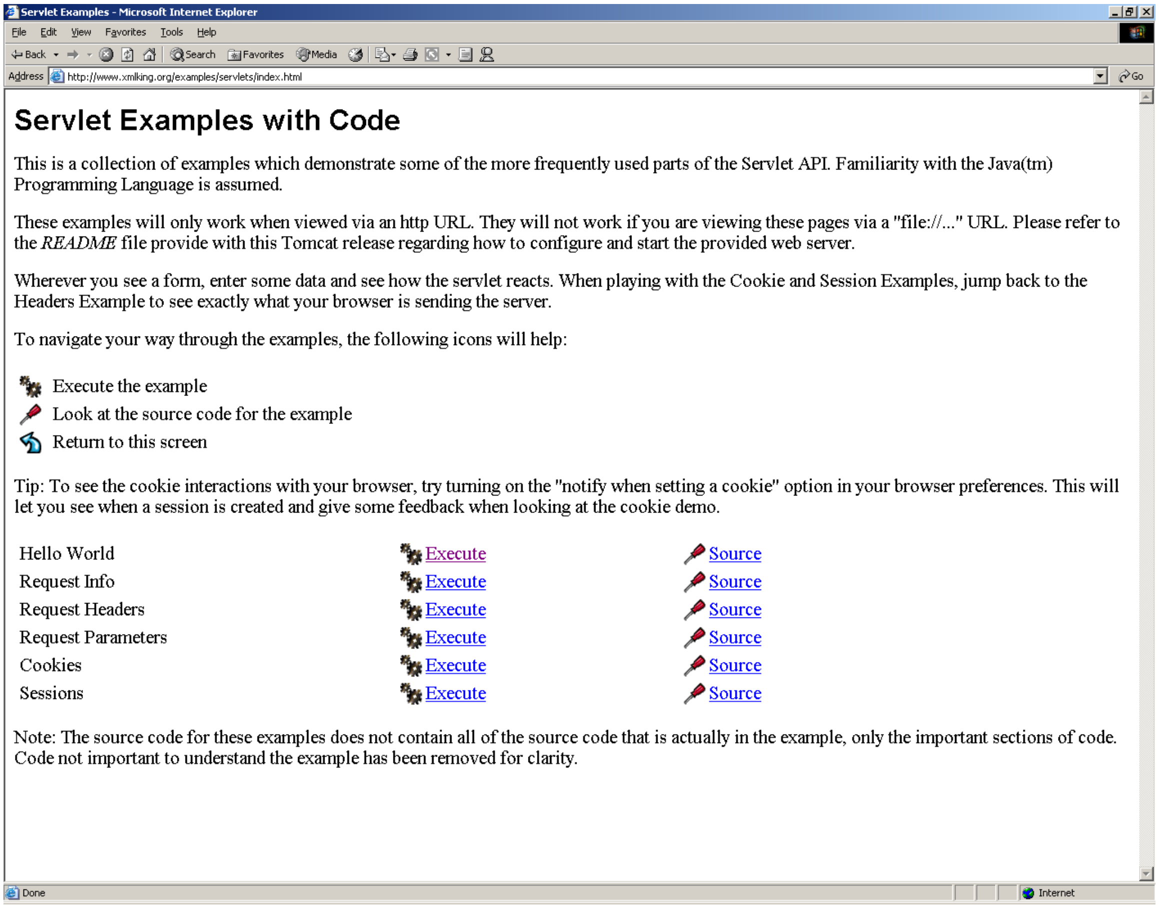
Figure B.9 DOS Window Tomcat startup.bat

6. DOS Window should OPEN displaying: figure B.9. Minimize this screen while Tomcat Servlet Engine is running on the server.



```
2001-10-11 09:22:08 - ContextManager: Adding context Ctx( /cocoon )
2001-10-11 09:22:08 - ContextManager: Adding context Ctx( /examples )
2001-10-11 09:22:08 - ContextManager: Adding context Ctx( /admin )
Starting tomcat. Check logs/tomcat.log for error messages
2001-10-11 09:22:08 - ContextManager: Adding context Ctx(  )
2001-10-11 09:22:08 - ContextManager: Adding context Ctx( /test )
2001-10-11 09:22:08 - PoolTcpConnector: Starting HttpConnectionHandler on 8080
2001-10-11 09:22:08 - PoolTcpConnector: Starting Ajp12ConnectionHandler on 8007
2001-10-11 09:22:08 - PoolTcpConnector: Starting Ajp13ConnectionHandler on 8009
```

Figure B.10 Tomcat Servlet Engine running


Servlet Examples - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www.xmlking.org/examples/servlets/index.html
Go
Done
Internet

Servlet Examples with Code

This is a collection of examples which demonstrate some of the more frequently used parts of the Servlet API. Familiarity with the Java(tm) Programming Language is assumed.

These examples will only work when viewed via an http URL. They will not work if you are viewing these pages via a "file://..." URL. Please refer to the *README* file provide with this Tomcat release regarding how to configure and start the provided web server.

Wherever you see a form, enter some data and see how the servlet reacts. When playing with the Cookie and Session Examples, jump back to the Headers Example to see exactly what your browser is sending the server.

To navigate your way through the examples, the following icons will help:

Execute the example
Look at the source code for the example
Return to this screen

Tip: To see the cookie interactions with your browser, try turning on the "notify when setting a cookie" option in your browser preferences. This will let you see when a session is created and give some feedback when looking at the cookie demo.

Hello World	Execute	Source
Request Info	Execute	Source
Request Headers	Execute	Source
Request Parameters	Execute	Source
Cookies	Execute	Source
Sessions	Execute	Source

Note: The source code for these examples does not contain all of the source code that is actually in the example, only the important sections of code. Code not important to understand the example has been removed for clarity.

Figure B.11 Tomcat Servlet Engine running

The user can test the stand-alone system by type http://localhost:8080/examples/Servlets/index.html on a browser with results displayed in figure B.10.

G. CONFIGURING TOMCAT SERVLET ENGINE TO OPERATE WITH THE APACHE WEB SERVER

The latest version mod_jk plug-in is needed to handle the communications between Tomcat and Apache and can be obtained at download directory at http://jakarta.apache.org/builds/jakarta-tomcat/release/v3.2-m1/bin/. The mod_jk can be obtained in two formats - binary and source. Depending on the platform, a binary version of mod_jk.dll is available for Windows 2000, recommended using the binary version. The version of mod_jk is not dependent on the version of Tomcat. The Tomcat 3.3 distribution of mod_jk will function correctly with other 3.x versions of Tomcat

If the binary is not available, follow the instructions for building mod_jk in Section E of this appendix or copy the binary file mod_jk.dll to the APACHE_HOME/libexec directory and skip Section E. The Source code is downloaded during installation of Tomcat. The source for mod_jk is included in the Binary Distribution of Tomcat in the TOMCAT_HOME/native/mod_jk/ directory. This directory is organized by Web Server name and version. Each directory contains the source as well as the appropriate build scripts, make files, or project files.

H. BUILDING MOD_JK FOR WINDOWS

The redirector was developed using Visual C++ version 6.0, so having this environment is a prerequisite if user needs to perform a custom build.

The steps Build mod_jk are:

1. Change directory to the Apache 1.3 or Apache 2.0 source directory depending on the version of Apache.

2. Set an APACHE1_HOME environment variable, which points to where your Apache is installed.

3. Execute the following command:

```
MSDEV mod_jk.dsp /MAKE ALL
```

If msdev is not in your path, enter the full path to msdev.exe. In addition, ApacheCore.lib is expected to exist in the APACHE1_HOME\src\CoreD and APACHE1_HOME\src\CoreR directories before linking will succeed. You will need to build enough of the Apache source to create these libraries.

4. Copy mod_jk.dll to Apache's modules directory.

This will build both release and debug versions of the redirector plug-in (mod_jk). An alternative will be to open mod_jk.dsp in msdev and build it using the build menu.

5. Copy mod_jk.dll to the APACHE_HOME/libexec directory.

I. CONFIGURING APACHE

This section details the configuration that is required for the Apache Web Server to support mod_jk and recommends removing all mod_jserv directives. Previously configure Apache http Web servers use mod_jserv, remove any ApJServMount directives from your httpd.conf. Remove old tomcat-apache.conf or tomcat.conf. The mod_jserv configuration directives are not compatible with mod_jk.

The simplest way to configure Apache to use mod_jk is to turn on the Apache auto-configure setting in Tomcat and put the following include directive at the end of your Apache httpd.conf file:

1. Select Start|Programs|Apache httpd Server|Configure Apache Server

2. Click on :Edit the Apache httpd.conf Configuration File.

3 . Type: Include C:/jakarta-tomcat-3.2.3/conf/MODJK.conf

```
#
# This comment was add by Clifton Williams to
# show the configuration changes made to HTTP.CONF to
# load the Coocon Web Framework, This include was add
# below the last closing directory tag </Directory>
#
Include C:/jakarta-tomcat-3.2.3/conf/MODJK.conf
```

Figure B.12 Apache httpd.conf Configuration File

This will tell Apache to use directives in the mod_jk.conf file in the Apache configuration.

J. CUSTOM CONFIGURATION OF TOMCAT MOD_JK.CONF

This an important section to understand because Cocoon use custom configuration to enable all cocoon function. The user can create custom configurations. This is done by enabling the auto-configuration and copying and renaming the TOMCAT_HOME/conf/auto/mod_jk.conf file a new configuration file, such as TOMCAT_HOME/conf/auto/mod_jk.conf-local.

Tomcat basic configuration:

The system needs to instruct Apache how to load Tomcat. This can be done with Apache's LoadModule and AddModule configuration directives. The mod_jk file informs the Web server of the location of workers.properties file. Mod_jk's uses JkWorkersFile configuration directive.

The user should specify a location of where mod_jk is going to place log file and a log level to be used. This is done using the JkLogFile and JkLogLevel configuration directives. Possible log levels are *debug*, *info*, *error* and *emerg*, but *info* should be in the user’s default selection. The directive JkLogStampFormat will configure the date/time format found on mod_jk logfile.

A simple example would be to include the following lines in the Apache httpd.conf file:

```
LoadModule    jk_module  libexec/mod_jk.so

AddModule     mod_jk.c

JkWorkersFile c:/jakarta-tomcat-3.2.3/conf/workers.properties

JkLogFile     c:/jakarta-tomcat-3.2.3/logs/mod_jk.log

JkLogLevel    info

JkLogStampFormat "[%a %b %d %H:%M:%S %Y] "
```

Assigning URLs to Tomcat for custom or local version of Mod_jk.conf file, the user can change settings such as the workers or URL prefix. Use mod_jk's JkMount directive to assign specific URLs to Tomcat. In general the structure of a JkMount directive is:

```
JkMount <URL prefix> <Worker name>
```

For example the following directives will send all requests ending in .jsp or beginning with /Servlet to the "ajp13" worker, but jsp requests to files located in /otherworker will go to "remoteworker".

```
JkMount /*.jsp ajp13

JkMount /Servlet/* ajp13

JkMount /otherworker/*.jsp remoteworker
```

The system can be configured to use the JkMount directive at the top level or inside <VirtualHost> sections of your httpd.conf file.

K. CONFIGURING TOMCAT

In most simple cases, Tomcat can generate the needed Apache configuration. The system can be configured so that when TOMCAT started up it will automatically generate a configuration file for Apache to use mod_jk. Most of the time, nothing is needed to do change except adding the following statement to TOMCAT_HOME/conf/server.xml file after

```
<AutoWebApp ... />.
    <ApacheConfig />
```

That's it, you can now start Tomcat and Apache and access Tomcat from the Apache server.

Note: Settings for mod_jk auto-configuration is new in Tomcat 3.3-b1. For Tomcat 3.2.3 and older versions of Tomcat, the file is created the auto-config file without have to add a directive in server.xml.

Tomcat 3.2.3 server.xml automatically adds new content information and updates TOMCAT_HOME/conf/jk/mod_jk.conf-auto. Later in this guide the user, generate a custom configuration mod_jk.conf-auto file for cocoon operations. This file provides context mounting URL prefixes that are not the default, user can use this file as a base for customized configurations and save the results in another file. If the system user needs to manage the Apache configuration, this file will need to update it whenever a new context is added.

Note: The user must restart Tomcat and Apache after adding a new context; Apache doesn't support configuration changes without a restart. In addition, the file TOMCAT_HOME/conf/jk/mod_jk.conf-auto is generated when Tomcat starts, so user needs to start Tomcat before Apache. Tomcat will overwrite TOMCAT_HOME/conf/jk/mod_jk.conf-auto each startup so customized configuration should be kept elsewhere. For example, copy TOMCAT_HOME/conf/jk/mod_jk.conf-auto to TOMCAT_HOME/conf/jk/mod_jk.conf-local before making changes. Tomcat need to be started to generate the configuration file after the initial load.

L. INSTALLING COCOON 1.8.2

The Cocoon program can be obtained at download directory at http://jakarta.apache.org/downloads. Cocoon can be downloaded and expanded. The source and binary code are available for distribution. Cocoon can be built using ANT and/or the binary code can be placed in the CLASSPATH for program development. The latest version Cocoon 1.8.2.zip is available online and is self-extracting.

This procedure will assist individuals in loading the binary code. The steps provided in building Ant would assist the user in building Cocoon from java source code into binary code for development. To make Cocoon work with Tomcat, the user must add a context to Tomcat that describes to Tomcat how to load Cocoon files. Then the user must tell Apache to send certain requests to Tomcat (and consequently Cocoon). Finally, the individual must provide the .XML files to be served by Cocoon.

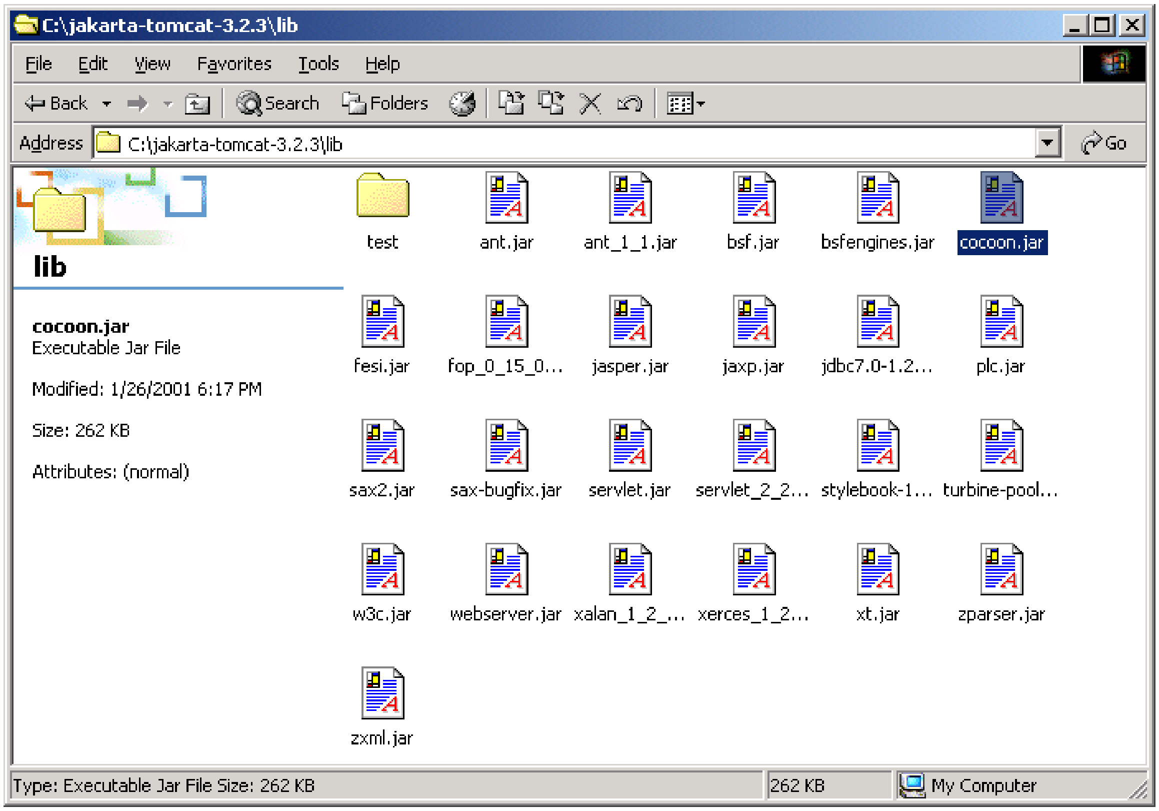
The first thing to do is to make sure that Cocoon and all the needed components (as explained in the previous section) are visible to the JVM. This means adding a bunch of jar package to the lib in Tomcat 3.2.3. The context in Tomcat describes to Tomcat how and when to load a particular Servlet and Cocoon is one such Servlet. Tomcat knows how to load the Cocoon .jar files. To begin with copy any .jar files from $COCOON_HOME/lib to $TOMCAT_HOME/lib that are necessary for Cocoon to run table B.2 show which jar files are required

Jar filename	**Package name**	**Obtained from**	**Purpose**	**Needed at run-time?**	**Needed at build-time?**	**Notes**
ant_1_1.jar	Ant 1.1	jakarta.apache.org	Cocoon build system (a bit like Make)	No	Yes	Other versions of Ant may not work.
bsfengines.jar	Bean Scripting Framework	IBM	Xalan extension functions	Optional	No	
bsf.jar	Bean Scripting Framework	IBM	Xalan extension functions	Optional	No	

fop_0_15_0.jar	FOP	xml.apache.org	Converts xsl:fo into PDF output	Yes (unless disabled)	Yes	
Sax-bugfix.jar	-	Robin Green	Fixes error reporting bug	Optional	No	See below
Servlet_2_2.jar	Servlet 2.2 API	jakarta.apache.org	Compiling	No	Yes, always	
Stylebook-1.0-b2.jar	Stylebook	xml.apache.org CVS	Generating docs	No	Yes for docs and dist	Requires Java 2; will be obsoleted by C2
Turbine-pool.jar	Turbine	java.apache.org	JDBC connection pooling	Yes, always	Yes	
W3c.jar	?	xml.apache.org	Required by FOP	Yes	?	
xalan_1_2_D02.jar	Xalan 1.2D02	xml.apache.org	Processing XSLT stylesheets and logicsheets	Yes	Yes	
Xerces_1_2.jar	Xerces 1.2	xml.apache.org	Parsing XML and outputting XML/HTML/text.	Yes	Yes	Position in CLASSPATH is very important (seebelow)

Table B.2 Cocoon jar files requirements

In addition, you must copy C:/Cocoon1.8.2/bin/cocoon.jar to C:/Jarakata-Tomcat-3.2.3/lib. Tomcat version 3.2.3 will automatically detect any .jar files in the C:/Jarakata-Tomcat-3.2.3/lib directory as in figure B.13.



cocoon.jar
Executable Jar File

Modified: 1/26/2001 6:17 PM

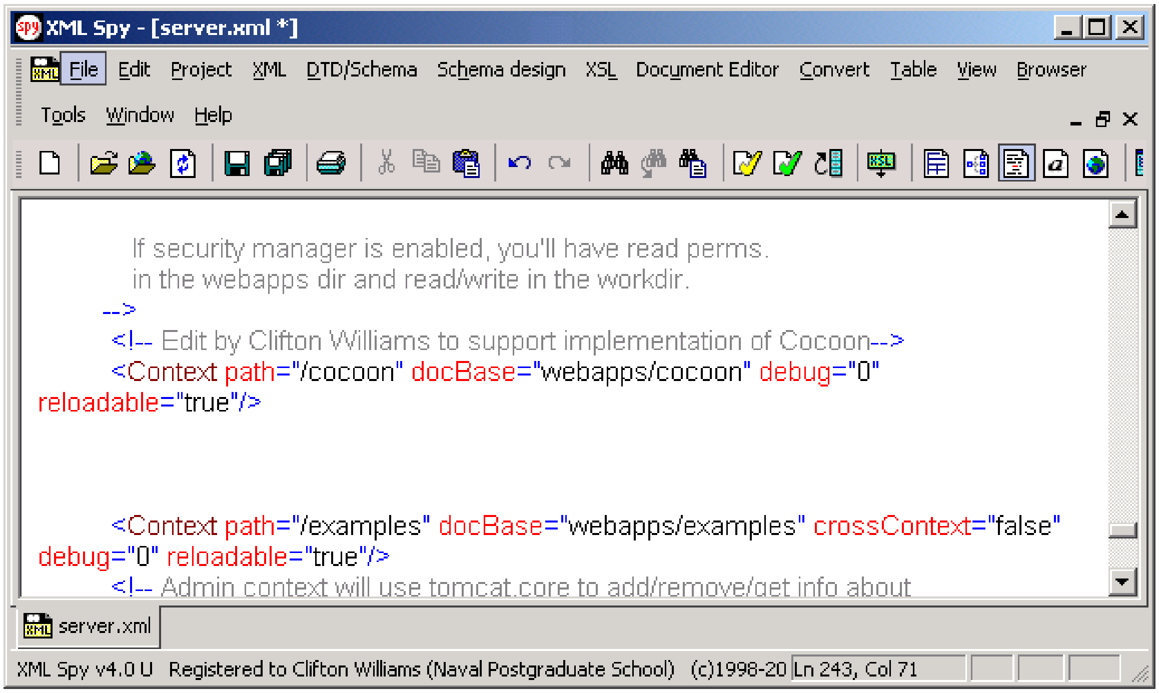
Size: 262 KB

Attributes: (normal)

Type: Executable Jar File Size: 262 KB | 262 KB

Figure B.13 Windows Display of C:/Jarakata-Tomcat-3.2.3/lib

Next Tomcat will need about the new context to run Cocoon requests. To do this edit the file C:/Jarakata-Tomcat-3.2.3/conf/server.xml and add the following line in figure B.14:


XML Spy - [server.xml *]
File Edit Project XML DTD/Schema Schema design XSL Document Editor Convert Table View Browser
Tools Window Help
server.xml
XML Spy v4.0 U Registered to Clifton Williams (Naval Postgraduate School) (c)1998-20 Ln 243, Col 71

```
            If security manager is enabled, you'll have read perms.
            in the webapps dir and read/write in the workdir.
        -->
         <!-- Edit by Clifton Williams to support implementation of Cocoon-->
         <Context path="/cocoon" docBase="webapps/cocoon" debug="0"
reloadable="true"/>



         <Context path="/examples" docBase="webapps/examples" crossContext="false"
debug="0" reloadable="true"/>
         <!-- Admin context will use tomcat.core to add/remove/get info about
```

Figure B.14 Server.XML file

This tells Tomcat that requests that come in under that partial path "/cocoon" should be mapped to the context defined in the directory "Webapps/cocoon".

Next, The user needs to tell Apache to forward the same partial pathnames to Tomcat. This is done using a custom configuration mod_jk.conf-auto file to enable all cocoon function.

Mod_jk.conf-auto file is regenerated and overwritten whenever Tomcat is started, so don't edit them directly, but instead save to a Cocoon.conf file. This tells Apache to direct .XML requests that come in under that partial path "/cocoon" to the directory under Tomcat C:/Jarakata-Tomcat-3.2.3/conf/Webapps/cocoon/WEB-INF. Configure Cocoon to use the newer Ajpv13 protocol. Both protocols are enabled by default. The "Ajp13" Connection Handler in Tomcat is a faster protocol and the ability to identify requests made via HTTPS. Edit Cocoon.conf to ensure the system contains the correct Ajpv13 protocol using figure B.15.

```
#########################################################
# Auto configuration for the /cocoon context starts.
#########################################################
#
# The following line makes apache aware of the location of the #/cocoon context
#
Alias /cocoon "C:/jakarta-tomcat-3.2.3/Webapps/cocoon"
<Directory "C:/jakarta-tomcat-3.2.3/Webapps/cocoon">
    Options Indexes FollowSymLinks
</Directory>
#
# The following line mounts all xml files and the /cocoon/ uri to tomcat
#
JkMount /cocoon/*.xml ajp13
AddType text/xml .xml
#
# The following line prohibits users from directly accessing WEB-INF
#
<Location "/cocoon/WEB-INF/">
    AllowOverride None
    deny from all
</Location>
#
# Use Directory too. On Windows, Location doesn't work unless case matches
```

```
#
<Directory "C:/jakarta-tomcat-3.2.3/Webapps/cocoon/WEB-INF/">
    AllowOverride None
    deny from all
</Directory>
#######################################################
# Auto configuration for the /cocoon context ends
```

Figure B.15 Cocoon.conf File

These files don't actually do anything by themselves. To be activated, a statement need to be included into the main Apache configuration file. To do this, add this to httpd.conf. The simplest way to configure Apache to use cocoon.conf is to turn on the Apache auto-configure setting in Tomcat and put the following include directive at the end of your Apache httpd.conf file:

1. Select Start|Programs|Apache httpd Server|Configure Apache Server
2. Click on :Edit the Apache httpd.conf Configuration File.
3 . Type: Include C:/jakarta-tomcat-3.2.3/conf/cocoon.conf

```
#
# This comment was add by Clifton Williams to
# show the configuration changes made to HTTP.CONF to
# load the Cocoon Web Framework, This include was add
# below the last closing directory tag </Directory>
#
# This is a custom configuration file used by Apache
# Tomcat and cocoon
```

```
Include C:/jakarta-tomcat-3.2.3/conf/cocoon.conf
```

Figure B.16 Apache httpd.conf Configuration File

This will tell Apache to use directives in the cocoon.conf file in the Apache configuration.

Finally, we need to set up the actual context that we have defined and pointed requests to above. To do this, we need to create a new directory in Webapps called cocoon. Then we need to make a sub-directory that describes to Tomcat how to map particular files to Cocoon, then we need to fill the sub-directory with our Cocoon source files (.XML files).

First, make a directory and its subdirectory:

1. Using Windows Explorer change directory to

C:/jakarta-tomcat-3.2.3/Webapps/

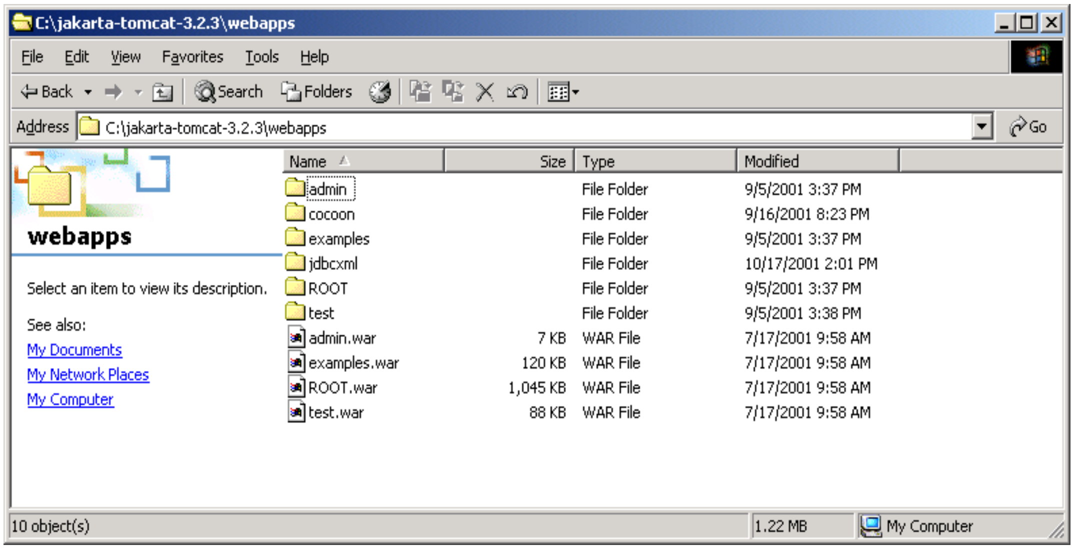
2. Select File|New|Folder

3. Name new folder “cocoon”

4. Left Click on cocoon

5. Select File|New|Folder

6. Name new folder “WEB-INF”



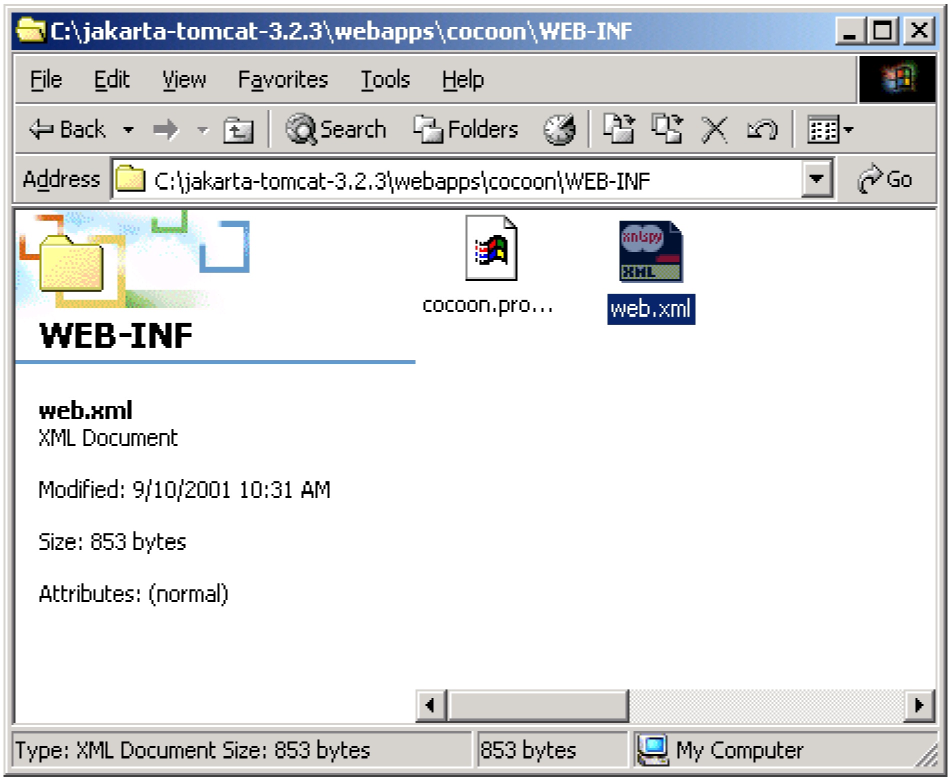
Name	Size	Type	Modified
admin		File Folder	9/5/2001 3:37 PM
cocoon		File Folder	9/16/2001 8:23 PM
examples		File Folder	9/5/2001 3:37 PM
jdbcxml		File Folder	10/17/2001 2:01 PM
ROOT		File Folder	9/5/2001 3:37 PM
test		File Folder	9/5/2001 3:38 PM
admin.war	7 KB	WAR File	7/17/2001 9:58 AM
examples.war	120 KB	WAR File	7/17/2001 9:58 AM
ROOT.war	1,045 KB	WAR File	7/17/2001 9:58 AM
test.war	88 KB	WAR File	7/17/2001 9:58 AM

Figure B.17   File Directory Structure Tomcat with Cocoon

Next, copy the template files from the Cocoon distribution:

Copy "Cocoon1.8.2/src/WEB-INF/Web.xml" file to "C:/jakarta-tomcat-3.2.3/Webapps/cocoon/WEB-INF" directory.

Copy "Cocoon1.8.2/conf/cocoon.properties" file to "C:/jakarta-tomcat-3.2.3/Webapps/cocoon/WEB-INF" directory.



C:\jakarta-tomcat-3.2.3\webapps\cocoon\WEB-INF
File Edit View Favorites Tools Help
Back
Search
Folders
Address
Go
WEB-INF
cocoon.pro...
web.xml

C:\jakarta-tomcat-3.2.3\webapps\cocoon\WEB-INF

web.xml
XML Document

Modified: 9/10/2001 10:31 AM

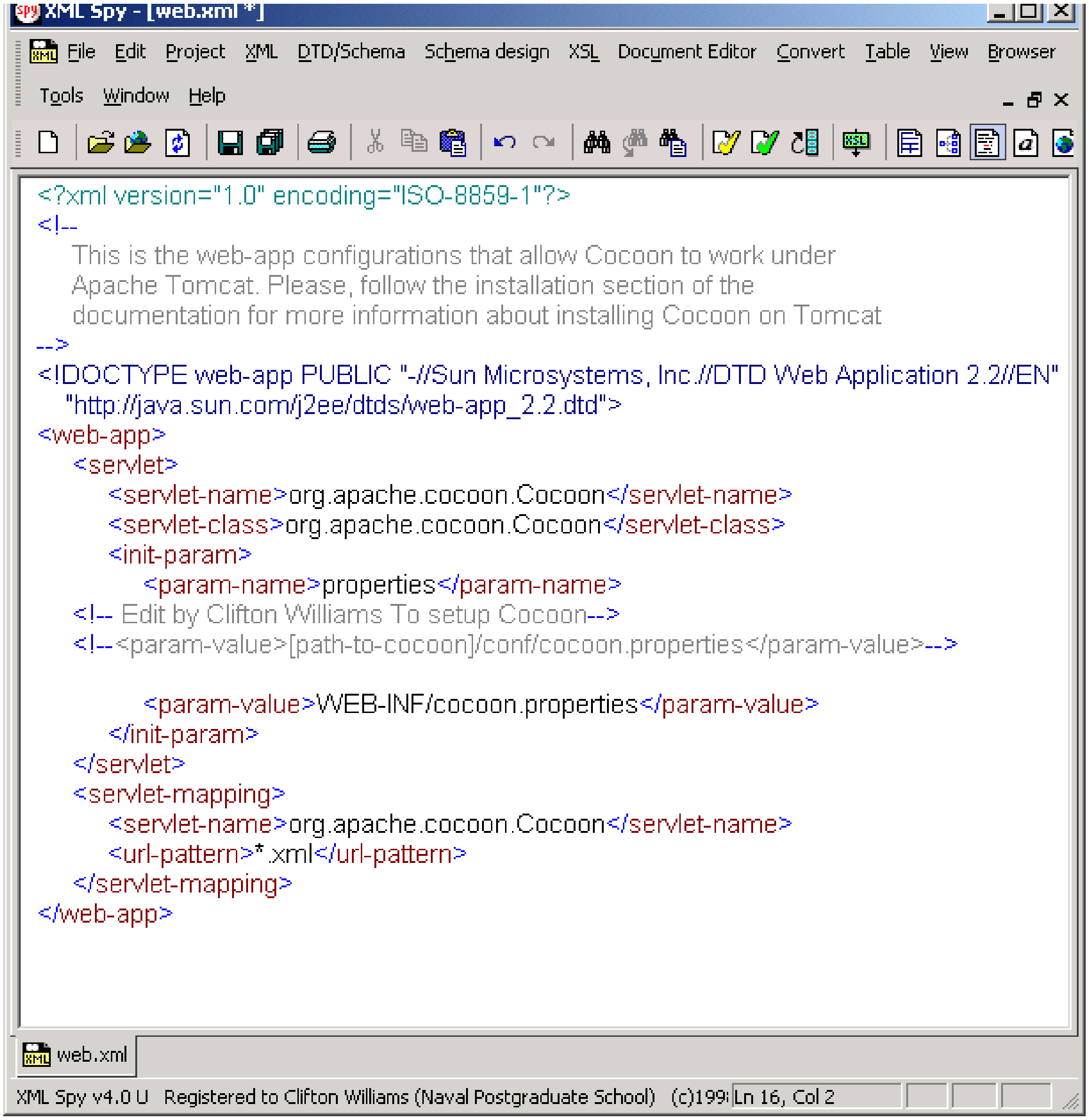
Size: 853 bytes

Attributes: (normal)

Type: XML Document Size: 853 bytes | 853 bytes | My Computer

Figure B.18 Directory Location Web.xml

The user needs edit the C:/jakarta-tomcat-3.2.3/Webapps/cocoon/WEB-INF/Web.xml file (Figure B.18 and Figure B.19) to point to the Cocoon properties file in the same directory. This by changing the text conf/cocoon.properties to WEB-INF/cocoon.properties.

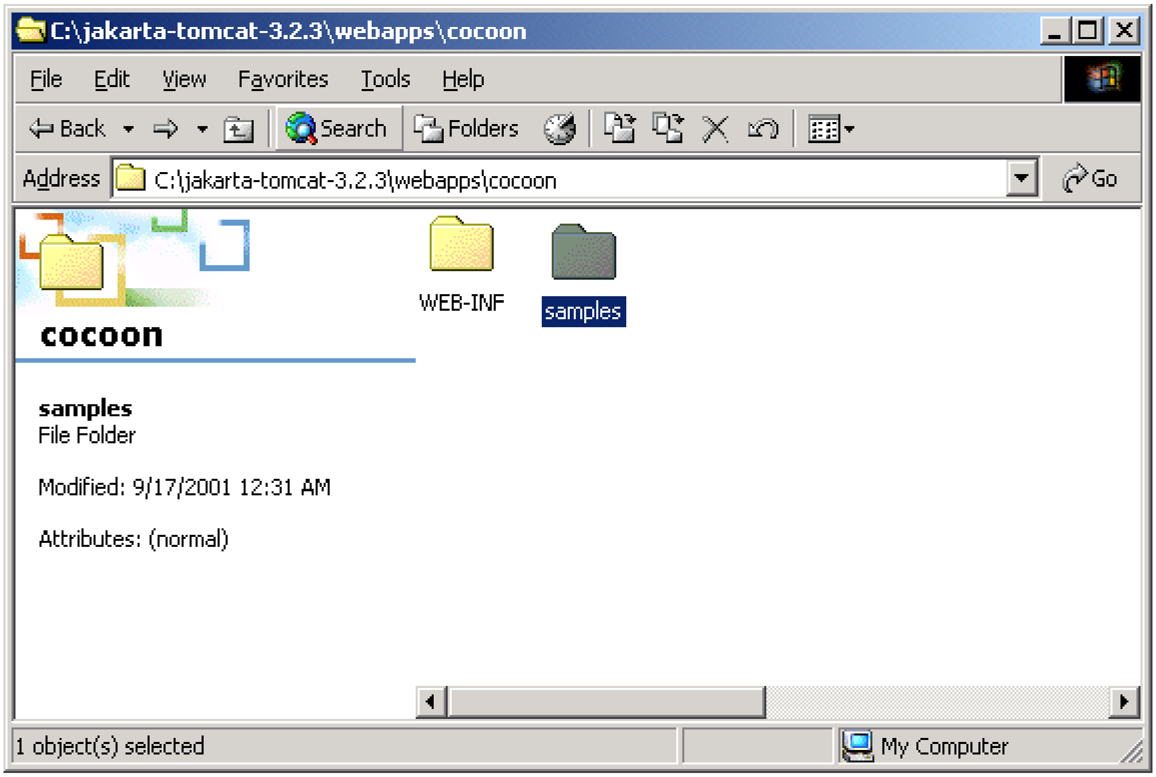


```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
   This is the web-app configurations that allow Cocoon to work under
   Apache Tomcat. Please, follow the installation section of the
   documentation for more information about installing Cocoon on Tomcat
-->
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
   "http://java.sun.com/j2ee/dtds/web-app_2.2.dtd">
<web-app>
   <servlet>
      <servlet-name>org.apache.cocoon.Cocoon</servlet-name>
      <servlet-class>org.apache.cocoon.Cocoon</servlet-class>
      <init-param>
         <param-name>properties</param-name>
   <!-- Edit by Clifton Williams To setup Cocoon-->
   <!--<param-value>[path-to-cocoon]/conf/cocoon.properties</param-value>-->

         <param-value>WEB-INF/cocoon.properties</param-value>
      </init-param>
   </servlet>
   <servlet-mapping>
      <servlet-name>org.apache.cocoon.Cocoon</servlet-name>
      <url-pattern>*.xml</url-pattern>
   </servlet-mapping>
</web-app>
```

Figure B.19 Web.xml Source Code

Note: that this path is a relative path, and must be so. Don't try to use an absolute path here. It won't work. Also, note that the Web.xml file describes how to map .XML requests to the Cocoon Servlet.

Populate the cocoon context with source .XML files. For testing purposes just use the samples that come along with Cocoon (Figure B.20). Note that some of these are insecure and SHOULD NOT be available on a production server, because they allow access to arbitrary .XML source code


C:\jakarta-tomcat-3.2.3\webapps\cocoon
File
Edit
View
Favorites
Tools
Help
Back
Search
Folders
Address
C:\jakarta-tomcat-3.2.3\webapps\cocoon
Go
cocoon
WEB-INF
samples

samples
File Folder

Modified: 9/17/2001 12:31 AM

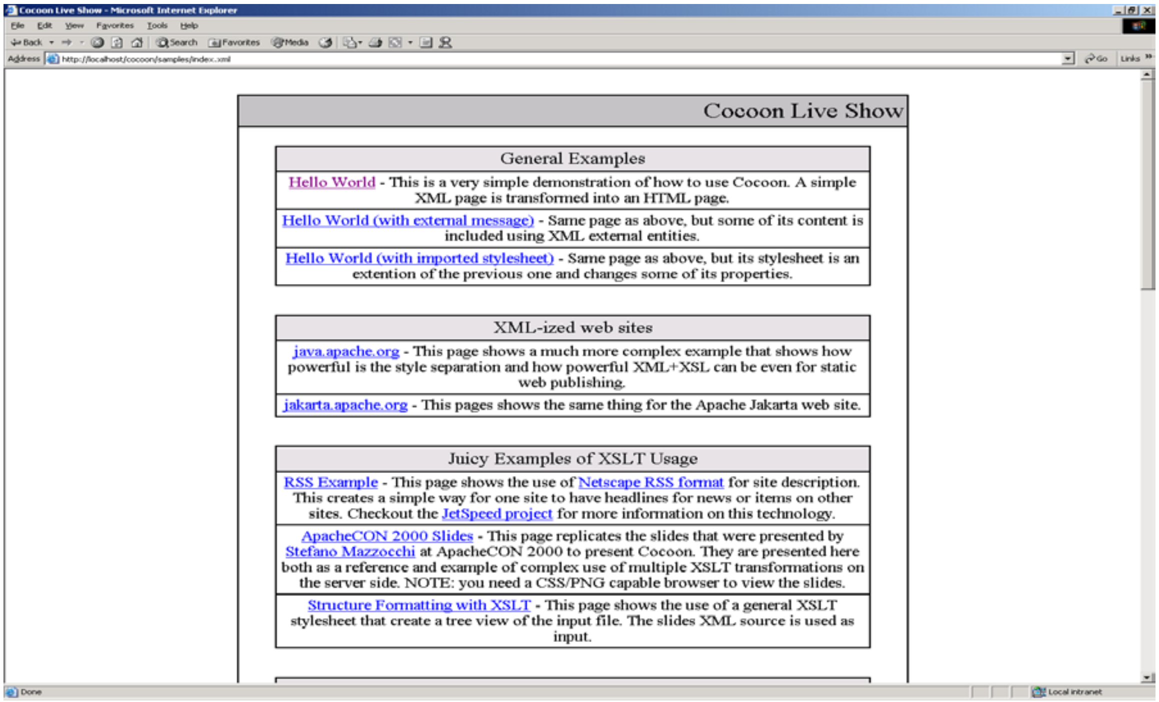
Attributes: (normal)

1 object(s) selected

My Computer

Figure B.20 Directory Location of Cocoon Samples

Finally, you need to stop Tomcat, stop Apache, and then restart the two in order to make the entire new settings load. You should be able to access pages like http://localhost/cocoon/samples/index.xml (Figure B.21).


Cocoon Live Show - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address http://localhost/cocoon/samples/index.xml
Go
Links
Done
Local intranet

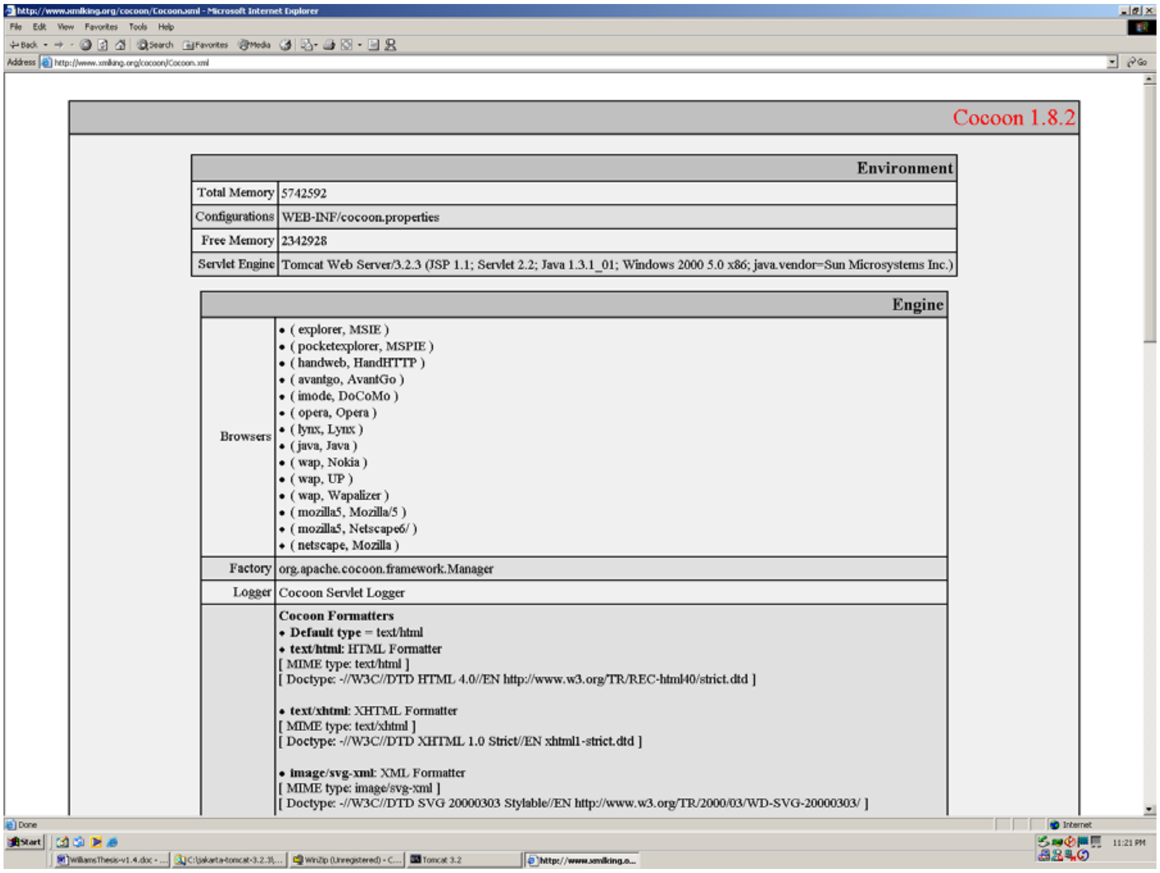
Cocoon Live Show

General Examples
Hello World - This is a very simple demonstration of how to use Cocoon. A simple XML page is transformed into an HTML page.
Hello World (with external message) - Same page as above, but some of its content is included using XML external entities.
Hello World (with imported stylesheet) - Same page as above, but its stylesheet is an extention of the previous one and changes some of its properties.

XML-ized web sites
java.apache.org - This page shows a much more complex example that shows how powerful is the style separation and how powerful XML+XSL can be even for static web publishing.
jakarta.apache.org - This pages shows the same thing for the Apache Jakarta web site.

Juicy Examples of XSLT Usage
RSS Example - This page shows the use of Netscape RSS format for site description. This creates a simple way for one site to have headlines for news or items on other sites. Checkout the JetSpeed project for more information on this technology.
ApacheCON 2000 Slides - This page replicates the slides that were presented by Stefano Mazzocchi at ApacheCON 2000 to present Cocoon. They are presented here both as a reference and example of complex use of multiple XSLT transformations on the server side. NOTE: you need a CSS/PNG capable browser to view the slides.
Structure Formatting with XSLT - This page shows the use of a general XSLT stylesheet that create a tree view of the input file. The slides XML source is used as input.

Figure B.21 Cocoon Index.xml Display


http://www.xmlking.org/cocoon/Cocoon.xml - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www.xmlking.org/cocoon/Cocoon.xml
Go
Done
Internet
Start
WilliamsThesis-v1.4.doc - ...
C:\jakarta-tomcat-3.2.3\...
WinZip (Unregistered) - C...
Tomcat 3.2
http://www.xmlking.o...
11:21 PM

Cocoon 1.8.2

	Environment
Total Memory	5742592
Configurations	WEB-INF/cocoon.properties
Free Memory	2342928
Servlet Engine	Tomcat Web Server/3.2.3 (JSP 1.1; Servlet 2.2; Java 1.3.1_01; Windows 2000 5.0 x86; java.vendor=Sun Microsystems Inc.)

	Engine
Browsers	• (explorer, MSIE) • (pocketexplorer, MSPIE) • (handweb, HandHTTP) • (avantgo, AvantGo) • (imode, DoCoMo) • (opera, Opera) • (lynx, Lynx) • (java, Java) • (wap, Nokia) • (wap, UP) • (wap, Wapalizer) • (mozilla5, Mozilla/5) • (mozilla5, Netscape6/) • (netscape, Mozilla)
Factory	org.apache.cocoon.framework.Manager
Logger	Cocoon Servlet Logger
	Cocoon Formatters • **Default type** = text/html • **text/html:** HTML Formatter [MIME type: text/html] [Doctype: -//W3C//DTD HTML 4.0//EN http://www.w3.org/TR/REC-html40/strict.dtd] • **text/xhtml**: XHTML Formatter [MIME type: text/xhtml] [Doctype: -//W3C//DTD XHTML 1.0 Strict//EN xhtml1-strict.dtd] • **image/svg-xml**: XML Formatter [MIME type: image/svg-xml] [Doctype: -//W3C//DTD SVG 20000303 Stylable//EN http://www.w3.org/TR/2000/03/WD-SVG-20000303/]

Figure B.22 Cocoon.xml Display

IMPORTANT: NoSuchMethodError message may be receive during starting up, make sure that xerces.jar is located before other XML jars in the CLASSPATH, otherwise XSP will not work. Tomcat 3.2.3 constructs it's own CLASSPATH automatically and then add the environment CLASSPATH, so manually setting the CLASSPATH before running Tomcat doesn't work. Instead, rename xml.jar to zxml.jar and parser.jar to zparser.jar.

M. INSTALLING MICROSOFT XML PARSER (MSXML) 4.0 FOR INTERNET EXPLORER

Windows Internet Explorer needs a XML Parser for browser to display XML. Microsoft XML Parser (MSXML) 4.0 formerly called the Microsoft XML Parser. This version has a number of improvements compared to MSXML 3.0. MSXML supports the

World Wide Web (W3) Consortium final recommendation for XML Schema, with both DOM and SAX. Microsoft XML Parser (MSXML) 4.0 is self-extracting and loads automatically. MSXML may be downloaded at http://msdn.microsoft.com/xml .

N. INSTALLING VIRTUAL REALITY MODELING LANGUAGE (VRML) AND X3D-EDIT

Installation of a Virtual Reality Modeling Language (VRML) plugins for the user Web browsers is required to use Cocoon with VRML. These plugins can work with any combination of Netscape and Internet Explorer. Installation X3D-Edit is being used to develop and test the Extensible 3D (X3D) tag set for the next-generation Virtual Reality Modeling Language (VRML 200x). X3D-Edit also exercises various X3D graphics rendering and translation implementations. Both X3D-Edit and the VRML plugins may be obtain through the Web site at http://www.Web3d.org/TaskGroups/x3d/translation/README.X3D-Edit.htm. This Web site provides complete and detail installation directions.

O. INSTALLING ADOBE PORTABLE DOCUMENT FORMAT (PDF) READER

Installation of Adobe Portable Document Format (PDF) reader is required to be use the users Web browser. Cocoon can automatically generate PDF from XML files using XSL:FO. Adobe Portable Document Format (PDF) is the open de facto standard for electronic document distribution worldwide. Adobe PDF is a universal file format that preserves all the fonts, formatting, graphics, and color of any source document, regardless of the application and platform used to create it. Adobe Portable Document Format (PDF) reader is self-extracting and loads automatically. Reader may be downloaded at htttp://www.adobe.com/products/acrobat/

APPENDIX C. EXTENSIBLE HTML

A. INTRODUCTION

The framework's prototype site uses XHTML in the Web site is design. All HTML file were converted into a XML document. A DTD validated the XML files and applied a XHTML stylesheet to create the output file of XHTML. Section B and C are the original HTML browser output and HTML document. Section D, E, and F are the new XML document, the external DTD, and the XHTML browser output.

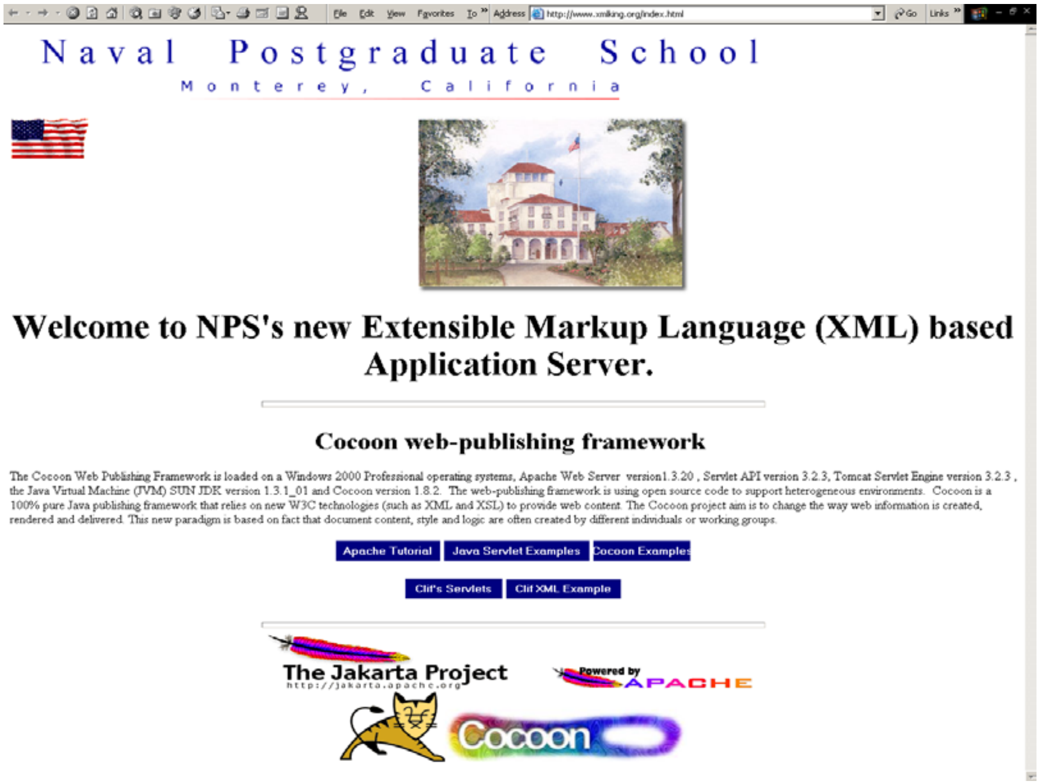
B. HMTL NPS XML SERVER HOME PAGE

```
<html>

<head>
<meta    http-equiv="Content-Type"    content="text/html;
charset=windows-1252">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>New Page 1</title>
</head>

<body>

<p><img                                          border="0"
src="file:///C:/Program%20Files/Apache%20Group/Apache/htdo
cs/images/NPS_r01_c1.gif" width="969" alt=
"NPS Header" height="90" align="middle" /></p>
<p><img                                          border="0"
src="file:///C:/Program%20Files/Apache%20Group/Apache/htdo
cs/images/Flag.gif" alt=
"Americam  Flag"  align="left"  width="100"  height="55"
/></p>
<p align="center"><img border="0" src=
"file:///C:/Program%20Files/Apache%20Group/Apache/htdocs/i
mages/HerrmannHallHomePage.jpg"  width="335"  height="232"
alt=
"Herrmann Hall" align="middle" /></p>

<h1 align="center">Welcome to NPS's new Extensible Markup
Language (XML) based Application Server. </h1>

<hr width="50%" size="8" />
<h2 align="center">Cocoon Web-publishing framework </h2>

<p class="MsoNormal" style=
'mso-layout-grid-align:none;text-autospace:none'><span
style=
"font-size:12.0pt;font-family:&quot;Times New Roman&quot;;
mso-fareast-font-family:&quot;Times New Roman&quot;;mso-
ansi-language:EN-US;mso-fareast-language: EN-US;mso-bidi-
language:AR-SA">
The Cocoon Web Publishing Framework is loaded on a Windows
2000
Professional operating systems, Apache Web Server
version1.3.20
, Servlet API version 3.2.3, Tomcat Servlet Engine version
3.2.3
, the Java Virtual Machine (JVM) SUN JDK version 1.3.1_01
and
Cocoon version 1.8.2. The Web-publishing framework is
using open
source code to support heterogeneous environments. Cocoon
is a
100% pure Java publishing framework that relies on new W3C
technologies (such as XML and XSL) to provide Web content.
The
Cocoon project aim is to change the way Web information is
created, rendered and delivered. This new paradigm is
based on
fact that document content, style and logic are often
created by
different individuals or working groups.</span></p>

<p align="center"><applet code="fphover.class"
codebase="file:///C:/Program%20Files/Apache%20Group/Apache
/htdocs/"
width="129" height="26"><param name="color"
value="#000080" />
<param name="hovercolor" value="#0000FF" />
<param name="textcolor" value="#FFFFFF" />
<param name="text" value="Apache Tutorial" />
<param name="effect" value="glow" />
<param name="url" valuetype="ref" value=
"http://www.xmlking.org/manual/misc/tutorials.html" />
</applet> <applet code="fphover.class"
codebase="file:///C:/Program%20Files/Apache%20Group/Apache
/htdocs/" width="180"
height="26"><param name="color" value="#000080" />
<param name="hovercolor" value="#0000FF" />
<param name="textcolor" value="#FFFFFF" />
<param name="text" value="Java Servlet Examples" />
<param name="effect" value="glow" />
<param name="url" valuetype="ref" value=
"http://www.xmlking.org/examples/Servlets/index.html" />
</applet> <applet code="fphover.class"
```

```
codebase="file:///C:/Program%20Files/Apache%20Group/Apache
/htdocs/" width="120"
height="26"><param name="color" value="#000080" />
<param name="hovercolor" value="#0000FF" />
<param name="textcolor" value="#FFFFFF" />
<param name="text" value="Cocoon Examples" />
<param name="effect" value="glow" />
<param name="url" valuetype="ref" value=
"http://www.xmlking.org/cocoon/samples/index.xml" />
</applet></p>

<p        align="center"><applet       code="fphover.class"
codebase="file:///C:/Program%20Files/Apache%20Group/Apache
/htdocs/"
width="120"        height="26"><param        name="color"
value="#000080" />
<param name="hovercolor" value="#0000FF" />
<param name="textcolor" value="#FFFFFF" />
<param name="text" value="Clif's Servlets" />
<param name="effect" value="glow" />
<param name="url" valuetype="ref" value=
"http://www.xmlking.org/Servlet.htm" />
</applet>           <applet          code="fphover.class"
codebase="file:///C:/Program%20Files/Apache%20Group/Apache
/htdocs/" width="142"
height="25"><param name="color" value="#000080" />
<param name="hovercolor" value="#0000FF" />
<param name="textcolor" value="#FFFFFF" />
<param name="effect" value="glow" />
<param name="text" value="Clif XML Example" />
<param name="url" valuetype="ref" value=
"http://www.xmlking.org/xmlcocoon.htm" />
</applet></p>

<hr width="50%" size="8" />
<div            align="center"><img            border="0"
src="file:///C:/Program%20Files/Apache%20Group/Apache/htdo
cs/images/jakarta-logo.gif"
alt="Jakarta"     width="350"     height="77"     /><img
src="file:///C:/Program%20Files/Apache%20Group/Apache/htdo
cs/apache_pb.gif"
alt="" width="259" height="38" /></div>

<div            align="center"><img            border="0"
src="file:///C:/Program%20Files/Apache%20Group/Apache/htdo
cs/images/tomcat.gif" alt=
"Tomact" width="130" height="92" /><img border="0" src=
"file:///C:/Program%20Files/Apache%20Group/Apache/htdocs/i
mages/cocoon.jpg"  alt="Cocoon"  width="300"  height="69"
/></div>
```

```
</body>

</html>
```

C. HTML BROWSER OUTPUT


File Edit View Favorites Tools
Address http://www.xmlking.org/index.html
Go
Links

Naval Postgraduate School

Monterey, California

Welcome to NPS's new Extensible Markup Language (XML) based Application Server.

Cocoon web-publishing framework

The Cocoon Web Publishing Framework is loaded on a Windows 2000 Professional operating systems, Apache Web Server version1.3.20 , Servlet API version 3.2.3, Tomcat Servlet Engine version 3.2.3 , the Java Virtual Machine (JVM) SUN JDK version 1.3.1_01 and Cocoon version 1.8.2. The web-publishing framework is using open source code to support heterogeneous environments. Cocoon is a 100% pure Java publishing framework that relies on new W3C technologies (such as XML and XSL) to provide web content. The Cocoon project aim is to change the way web information is created, rendered and delivered. This new paradigm is based on fact that document content, style and logic are often created by different individuals or working groups.

D. XHMTL NPS XML SERVER HOME PAGE

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>NPS XML Server Home Page</title>
</head>
<body bgcolor="#FFFFFF" text="#000000" link="#0000FF"
vlink=
"#000080" alink="#FF0000">
<p><img border="0" src="images/NPS_r01_c1.gif" width="969"
alt=
"NPS Header" height="90" align="middle" /></p>
```

```
<p align="left"><img border="0" src="images/Flag.gif" alt=
"Americam Flag" align="left" width="100" height="55" /></p>

<p align="center"><img border="0" src=
"images/HerrmannHallHomePage.jpg" width="335" height="232"
alt=
"Herrmann Hall" align="middle" /></p>

<h1 align="center">Welcome to NPS's new Extensible Markup
Language (XML) based Application Server. </h1>

<hr width="50%" size="8" />
<h2 align="center">Cocoon Web-publishing framework </h2>

<p class="MsoNormal" style=
'mso-layout-grid-align:none;text-autospace:none'><span
style=
"font-size:12.0pt;font-family:&quot;Times New Roman&quot;;
mso-fareast-font-family:&quot;Times New Roman&quot;;mso-
ansi-language:EN-US;mso-fareast-language: EN-US;mso-bidi-
language:AR-SA">
The Cocoon Web Publishing Framework is loaded on a Windows
2000
Professional operating systems, Apache Web Server
version1.3.20
, Servlet API version 3.2.3, Tomcat Servlet Engine version
3.2.3
, the Java Virtual Machine (JVM) SUN JDK version 1.3.1_01
and
Cocoon version 1.8.2.  The Web-publishing framework is
using open
source code to support heterogeneous environments.  Cocoon
is a
100% pure Java publishing framework that relies on new W3C
technologies (such as XML and XSL) to provide Web content.
The
Cocoon project aim is to change the way Web information is
created, rendered and delivered. This new paradigm is based
on
fact that document content, style and logic are often
created by
different individuals or working groups.</span></p>

<p align="center"><applet code="fphover.class"
codebase="./"
width="129" height="26"><param name="color" value="#000080"
/>
<param name="hovercolor" value="#0000FF" />
<param name="textcolor" value="#FFFFFF" />
<param name="text" value="Apache Tutorial" />
```

```
<param name="effect" value="glow" />
<param name="url" valuetype="ref" value=
"http://www.xmlking.org/manual/misc/tutorials.html" />
</applet> <applet code="fphover.class" codebase="./"
width="180"
height="26"><param name="color" value="#000080" />
<param name="hovercolor" value="#0000FF" />
<param name="textcolor" value="#FFFFFF" />
<param name="text" value="Java Servlet Examples" />
<param name="effect" value="glow" />
<param name="url" valuetype="ref" value=
"http://www.xmlking.org/examples/Servlets/index.html" />
</applet> <applet code="fphover.class" codebase="./"
width="120"
height="26"><param name="color" value="#000080" />
<param name="hovercolor" value="#0000FF" />
<param name="textcolor" value="#FFFFFF" />
<param name="text" value="Cocoon Examples" />
<param name="effect" value="glow" />
<param name="url" valuetype="ref" value=
"http://www.xmlking.org/cocoon/samples/index.xml" />
</applet></p>

<p align="center"><applet code="fphover.class"
codebase="./"
width="120" height="26"><param name="color" value="#000080"
/>
<param name="hovercolor" value="#0000FF" />
<param name="textcolor" value="#FFFFFF" />
<param name="text" value="Clif's Servlets" />
<param name="effect" value="glow" />
<param name="url" valuetype="ref" value=
"http://www.xmlking.org/Servlet.htm" />
</applet> <applet code="fphover.class" codebase="./"
width="142"
height="25"><param name="color" value="#000080" />
<param name="hovercolor" value="#0000FF" />
<param name="textcolor" value="#FFFFFF" />
<param name="effect" value="glow" />
<param name="text" value="Clif XML Example" />
<param name="url" valuetype="ref" value=
"http://www.xmlking.org/xmlcocoon.htm" />
</applet></p>

<hr width="50%" size="8" />
<div align="center"><img border="0" src="images/jakarta-
logo.gif"
alt="Jakarta" width="350" height="77" /><img
src="apache_pb.gif"
alt="" width="259" height="38" /></div>
```

```
<div align="center"><img border="0" src="images/tomcat.gif"
alt=
"Tomact" width="130" height="92" /><img border="0" src=
"images/cocoon.jpg" alt="Cocoon" width="300" height="69"
/></div>
</body>
</html>
```

E. XHTML TRANSITIONAL DTD

```
<!--
   Extensible HTML version 1.0 Transitional DTD

   This is the same as HTML 4.0 Transitional except for
   changes due to the differences between XML and SGML.

   Namespace = http://www.w3.org/1999/xhtml

   For further information, see: http://www.w3.org/TR/xhtml1

   Copyright (c) 1998-2000 W3C (MIT, INRIA, Keio),
   All Rights Reserved.

   This DTD module is identified by the PUBLIC and SYSTEM
identifiers:

   PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   SYSTEM "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd"

   $Revision: 1.1 $
   $Date: 2000/01/26 14:08:56 $

-->
<!--================ Character mnemonic entities
=========================-->
<!ENTITY % HTMLlat1 PUBLIC
   "-//W3C//ENTITIES Latin 1 for XHTML//EN"
   "xhtml-lat1.ent">
%HTMLlat1;
<!ENTITY % HTMLsymbol PUBLIC
   "-//W3C//ENTITIES Symbols for XHTML//EN"
   "xhtml-symbol.ent">
%HTMLsymbol;
<!ENTITY % HTMLspecial PUBLIC
   "-//W3C//ENTITIES Special for XHTML//EN"
   "xhtml-special.ent">
%HTMLspecial;
<!--================== Imported Names
====================================-->
<!ENTITY % ContentType "CDATA">
<!-- media type, as per [RFC2045] -->
```

```
<!ENTITY % ContentTypes "CDATA">
<!-- comma-separated list of media types, as per [RFC2045] -->
<!ENTITY % Charset "CDATA">
<!-- a character encoding, as per [RFC2045] -->
<!ENTITY % Charsets "CDATA">
<!-- a space separated list of character encodings, as per
[RFC2045] -->
<!ENTITY % LanguageCode "NMTOKEN">
<!-- a language code, as per [RFC1766] -->
<!ENTITY % Character "CDATA">
<!-- a single character from [ISO10646] -->
<!ENTITY % Number "CDATA">
<!-- one or more digits -->
<!ENTITY % LinkTypes "CDATA">
<!-- space-separated list of link types -->
<!ENTITY % MediaDesc "CDATA">
<!-- single or comma-separated list of media descriptors -->
<!ENTITY % URI "CDATA">
<!-- a Uniform Resource Identifier, see [RFC2396] -->
<!ENTITY % UriList "CDATA">
<!-- a space separated list of Uniform Resource Identifiers -->
<!ENTITY % Datetime "CDATA">
<!-- date and time information. ISO date format -->
<!ENTITY % Script "CDATA">
<!-- script expression -->
<!ENTITY % StyleSheet "CDATA">
<!-- style sheet data -->
<!ENTITY % Text "CDATA">-
<!-- used for titles etc. -->
<!ENTITY % FrameTarget "NMTOKEN">
<!-- render in this frame -->
<!ENTITY % Length "CDATA">
<!-- nn for pixels or nn% for percentage length -->
<!ENTITY % MultiLength "CDATA">
<!-- pixel, percentage, or relative -->
<!ENTITY % MultiLengths "CDATA">
<!-- comma-separated list of MultiLength -->
<!ENTITY % Pixels "CDATA">
<!-- integer representing length in pixels -->
<!-- these are used for image maps -->
<!ENTITY % Shape "(rect|circle|poly|default)">
<!ENTITY % Coords "CDATA">
<!-- comma separated list of lengths -->
<!-- used for object, applet, img, input and iframe -->
<!ENTITY % ImgAlign "(top|middle|bottom|left|right)">
<!-- a color using sRGB: #RRGGBB as Hex values -->
<!ENTITY % Color "CDATA">
<!-- There are also 16 widely known color names with their sRGB
values:

    Black  = #000000    Green  = #008000
    Silver = #C0C0C0    Lime   = #00FF00
    Gray   = #808080    Olive  = #808000
    White  = #FFFFFF    Yellow = #FFFF00
    Maroon = #800000    Navy   = #000080
    Red    = #FF0000    Blue   = #0000FF
    Purple = #800080    Teal   = #008080
    Fuchsia= #FF00FF    Aqua   = #00FFFF
-->
<!--===================                Generic             Attributes
===============================-->
<!-- core attributes common to most elements
  id       document-wide unique id
  class    space separated list of classes
  style    associated style info
  title    advisory title/amplification
-->
<!ENTITY % coreattrs "id          ID              #IMPLIED
  class       CDATA          #IMPLIED
  style       %StyleSheet;   #IMPLIED
  title       %Text;         #IMPLIED">
<!-- internationalization attributes
  lang        language code (backwards compatible)
  xml:lang    language code (as per XML 1.0 spec)
  dir         direction for weak/neutral text
-->
<!ENTITY % i18n "lang        %LanguageCode; #IMPLIED
  xml:lang    %LanguageCode; #IMPLIED
  dir         (ltr|rtl)      #IMPLIED">
<!-- attributes for common UI events
  onclick     a pointer button was clicked
  ondblclick  a pointer button was double clicked
  onmousedown a pointer button was pressed down
  onmouseup   a pointer button was released
  onmousemove a pointer was moved onto the element
  onmouseout  a pointer was moved away from the element
  onkeypress  a key was pressed and released
  onkeydown   a key was pressed down
  onkeyup     a key was released
-->
<!ENTITY % events "onclick     %Script;       #IMPLIED
  ondblclick  %Script;       #IMPLIED
  onmousedown %Script;       #IMPLIED
  onmouseup   %Script;       #IMPLIED
  onmouseover %Script;       #IMPLIED
  onmousemove %Script;       #IMPLIED
  onmouseout  %Script;       #IMPLIED
  onkeypress  %Script;       #IMPLIED
  onkeydown   %Script;       #IMPLIED
  onkeyup     %Script;       #IMPLIED">
<!-- attributes for elements that can get the focus
  accesskey   accessibility key character
  tabindex    position in tabbing order
  onfocus     the element got the focus
  onblur      the element lost the focus
-->
<!ENTITY % focus "accesskey   %Character;    #IMPLIED
  tabindex    %Number;       #IMPLIED
  onfocus     %Script;       #IMPLIED
  onblur      %Script;       #IMPLIED">
```

```
<!ENTITY % attrs "%coreattrs; %i18n; %events;">
<!-- text alignment for p, div, h1-h6. The default is
     align="left" for ltr headings, "right" for rtl -->
<!ENTITY % TextAlign "align (left|center|right) #IMPLIED">
<!--===================                Text             Elements
====================================-->
<!ENTITY % special "br | span | bdo | object | applet | img | map
| iframe">
<!ENTITY % fontstyle "tt | i | b | big | small | u
                      | s | strike |font | basefont">
<!ENTITY % phrase "em | strong | dfn | code | q | sub | sup |
                   samp | kbd | var | cite | abbr | acronym">
<!ENTITY  %  inline.forms  "input  |  select  |  textarea  |  label  |
button">
<!-- these can occur at block or inline level -->
<!ENTITY % misc "ins | del | script | noscript">
<!ENTITY  %  inline  "a  |  %special;  |  %fontstyle;  |  %phrase;  |
%inline.forms;">
<!-- %Inline; covers inline or "text-level" elements -->
<!ENTITY % Inline "(#PCDATA | %inline; | %misc;)*">
<!--==================        Block         level         elements
==============================-->
<!ENTITY % heading "h1|h2|h3|h4|h5|h6">
<!ENTITY % lists "ul | ol | dl | menu | dir">
<!ENTITY  % blocktext "pre | hr | blockquote | address | center |
noframes">
<!ENTITY  %  block  "p  |  %heading;  |  div  |  %lists;  |  %blocktext;  |
isindex |fieldset | table">
<!ENTITY % Block "(%block; | form | %misc;)*">
<!--  %Flow;  mixes  Block  and  Inline  and  is  used  for  list  items
etc. -->
<!ENTITY  %  Flow  "(#PCDATA  |  %block;  |  form  |  %inline;  |
%misc;)*">
<!--==================     Content     models     for     exclusions
=====================-->
<!-- a elements use %Inline; excluding a -->
<!ENTITY  %  a.content  "(#PCDATA  |  %special;  |  %fontstyle;  |
%phrase; | %inline.forms; | %misc;)*">
<!-- pre uses %Inline excluding img, object, applet, big, small,
     sub, sup, font, or basefont -->
<!ENTITY % pre.content "(#PCDATA | a | br | span | bdo | map | tt
| i | b | u | s |
      %phrase; | %inline.forms;)*">
<!-- form uses %Flow; excluding form -->
<!ENTITY  %  form.content  "(#PCDATA  |  %block;  |  %inline;  |
%misc;)*">
<!--  button  uses  %Flow;  but  excludes  a,  form,  form  controls,
iframe -->
<!ENTITY  %  button.content  "(#PCDATA  |  p  |  %heading;  |  div  |
%lists; | %blocktext; |
      table | br | span | bdo | object | applet | img | map |
      %fontstyle; | %phrase; | %misc;)*">
<!--================              Document              Structure
==================================-->
<!-- the namespace URI designates the document profile -->
```

```
<!ELEMENT html (head, body)>
<!ATTLIST html
  %i18n;
  xmlns %URI; #FIXED "http://www.w3.org/1999/xhtml"
>
<!--================                Document                Head
=======================================-->
<!ENTITY % head.misc "(script|style|meta|link|object|isindex)*">
<!-- content model is %head.misc; combined with a single
     title and an optional base element in any order -->
<!ELEMENT  head  (%head.misc;,  ((title,  %head.misc;,  (base,
%head.misc;)?) | (base, %head.misc;, (title, %head.misc;))))>
<!ATTLIST head
  %i18n;
  profile %URI; #IMPLIED
>
<!--  The  title  element  is  not  considered  part  of  the  flow  of
text.
       It should be displayed, for example as the page header or
       window title. Exactly one title is required per document.
    -->
<!ELEMENT title (#PCDATA)>
<!ATTLIST title
  %i18n;
>
<!-- document base URI -->
<!ELEMENT base EMPTY>
<!ATTLIST base
  href %URI; #IMPLIED
  target %FrameTarget; #IMPLIED
>
<!-- generic metainformation -->
<!ELEMENT meta EMPTY>
<!ATTLIST meta
  %i18n;
  http-equiv CDATA #IMPLIED
  name CDATA #IMPLIED
  content CDATA #REQUIRED
  scheme CDATA #IMPLIED
>
<!--
  Relationship values can be used in principle:

   a) for document specific toolbars/menus when used
      with the link element in document head e.g.
        start, contents, previous, next, index, end, help
   b) to link to a separate style sheet (rel="stylesheet")
   c) to make a link to a script (rel="script")
   d) by stylesheets to control how collections of
      html nodes are rendered into printed documents
   e) to make a link to a printable version of this document
      e.g.   a   PostScript   or   PDF   version   (rel="alternate"
media="print")
-->
<!ELEMENT link EMPTY>
```

```
<!ATTLIST link
   %attrs;
   charset %Charset; #IMPLIED
   href %URI; #IMPLIED
   hreflang %LanguageCode; #IMPLIED
   type %ContentType; #IMPLIED
   rel %LinkTypes; #IMPLIED
   rev %LinkTypes; #IMPLIED
   media %MediaDesc; #IMPLIED
   target %FrameTarget; #IMPLIED
>
<!-- style info, which may include CDATA sections -->
<!ELEMENT style (#PCDATA)>
<!ATTLIST style
   %i18n;
   type %ContentType; #REQUIRED
   media %MediaDesc; #IMPLIED
   title %Text; #IMPLIED
   xml:space (preserve) #FIXED "preserve"
>
<!-- script statements, which may include CDATA sections -->
<!ELEMENT script (#PCDATA)>
<!ATTLIST script
   charset %Charset; #IMPLIED
   type %ContentType; #REQUIRED
   language CDATA #IMPLIED
   src %URI; #IMPLIED
   defer (defer) #IMPLIED
   xml:space (preserve) #FIXED "preserve"
>
<!-- alternate content container for non script-based rendering -
->
<!ELEMENT noscript %Flow;>
<!ATTLIST noscript
   %attrs;
>
<!--=======================                                 Frames
=======================================-->
<!-- inline subwindow -->
<!ELEMENT iframe %Flow;>
<!ATTLIST iframe
   %coreattrs;
   longdesc %URI; #IMPLIED
   name NMTOKEN #IMPLIED
   src %URI; #IMPLIED
   frameborder (1 | 0) "1"
   marginwidth %Pixels; #IMPLIED
   marginheight %Pixels; #IMPLIED
   scrolling (yes | no | auto) "auto"
   align %ImgAlign; #IMPLIED
   height %Length; #IMPLIED
   width %Length; #IMPLIED
>
<!-- alternate content container for non frame-based rendering --
>
```

```
<!ELEMENT noframes %Flow;>
<!ATTLIST noframes
  %attrs;
>
<!--===================                Document                Body
====================================-->
<!ELEMENT body %Flow;>
<!ATTLIST body
  %attrs;
  onload %Script; #IMPLIED
  onunload %Script; #IMPLIED
  background %URI; #IMPLIED
  bgcolor %Color; #IMPLIED
  text %Color; #IMPLIED
  link %Color; #IMPLIED
  vlink %Color; #IMPLIED
  alink %Color; #IMPLIED
>
<!ELEMENT div %Flow;>
<!-- generic language/style container -->
<!ATTLIST div
  %attrs;
  %TextAlign;
>
<!--===================                                   Paragraphs
=======================================-->
<!ELEMENT p %Inline;>
<!ATTLIST p
  %attrs;
  %TextAlign;
>
<!--===================                                     Headings
=========================================-->
<!--
  There are six levels of headings from h1 (the most important)
  to h6 (the least important).
-->
<!ELEMENT h1 %Inline;>
<!ATTLIST h1
  %attrs;
  %TextAlign;
>
<!ELEMENT h2 %Inline;>
<!ATTLIST h2
  %attrs;
  %TextAlign;
>
<!ELEMENT h3 %Inline;>
<!ATTLIST h3
  %attrs;
  %TextAlign;
>
<!ELEMENT h4 %Inline;>
<!ATTLIST h4
  %attrs;
```

```
   %TextAlign;
>
<!ELEMENT h5 %Inline;>
<!ATTLIST h5
   %attrs;
   %TextAlign;
>
<!ELEMENT h6 %Inline;>
<!ATTLIST h6
   %attrs;
   %TextAlign;
>
<!--===================                                      Lists
============================================-->
<!-- Unordered list bullet styles -->
<!ENTITY % ULStyle "(disc|square|circle)">
<!-- Unordered list -->
<!ELEMENT ul (li)+>
<!ATTLIST ul
   %attrs;
   type %ULStyle; #IMPLIED
   compact (compact) #IMPLIED
>
<!-- Ordered list numbering style

    1   arabic numbers      1, 2, 3, ...
    a   lower alpha         a, b, c, ...
    A   upper alpha         A, B, C, ...
    i   lower roman         i, ii, iii, ...
    I   upper roman         I, II, III, ...

    The style is applied to the sequence number which by default
    is reset to 1 for the first list item in an ordered list.
-->
<!ENTITY % OLStyle "CDATA">
<!-- Ordered (numbered) list -->
<!ELEMENT ol (li)+>
<!ATTLIST ol
   %attrs;
   type %OLStyle; #IMPLIED
   compact (compact) #IMPLIED
   start %Number; #IMPLIED
>
<!-- single column list (DEPRECATED) -->
<!ELEMENT menu (li)+>
<!ATTLIST menu
   %attrs;
   compact (compact) #IMPLIED
>
<!-- multiple column list (DEPRECATED) -->
<!ELEMENT dir (li)+>
<!ATTLIST dir
   %attrs;
   compact (compact) #IMPLIED
>
```

```
<!-- LIStyle is constrained to: "(%ULStyle;|%OLStyle;)" -->
<!ENTITY % LIStyle "CDATA">
<!-- list item -->
<!ELEMENT li %Flow;>
<!ATTLIST li
  %attrs;
  type %LIStyle; #IMPLIED
  value %Number; #IMPLIED
>
<!-- definition lists - dt for term, dd for its definition -->
<!ELEMENT dl (dt | dd)+>
<!ATTLIST dl
  %attrs;
  compact (compact) #IMPLIED
>
<!ELEMENT dt %Inline;>
<!ATTLIST dt
  %attrs;
>
<!ELEMENT dd %Flow;>
<!ATTLIST dd
  %attrs;
>
<!--===================                                    Address
==========================================-->
<!-- information on author -->
<!ELEMENT address %Inline;>
<!ATTLIST address
  %attrs;
>
<!--===================               Horizontal              Rule
==================================-->
<!ELEMENT hr EMPTY>
<!ATTLIST hr
  %attrs;
  align (left | center | right) #IMPLIED
  noshade (noshade) #IMPLIED
  size %Pixels; #IMPLIED
  width %Length; #IMPLIED
>
<!--===================              Preformatted             Text
================================-->
<!-- content is %Inline; excluding
        "img|object|applet|big|small|sub|sup|font|basefont" -->
<!ELEMENT pre %pre.content;>
<!ATTLIST pre
  %attrs;
  width %Number; #IMPLIED
  xml:space (preserve) #FIXED "preserve"
>
<!--===================              Block-like             Quotes
================================-->
<!ELEMENT blockquote %Flow;>
<!ATTLIST blockquote
  %attrs;
```

```
   cite %URI; #IMPLIED
>
<!--=================== Text alignment
===================================-->
<!-- center content -->
<!ELEMENT center %Flow;>
<!ATTLIST center
   %attrs;
>
<!--=================== Inserted/Deleted Text
============================-->
<!--
  ins/del are allowed in block and inline content, but its
  inappropriate to include block content within an ins element
  occurring in inline content.
-->
<!ELEMENT ins %Flow;>
<!ATTLIST ins
   %attrs;
   cite %URI; #IMPLIED
   datetime %Datetime; #IMPLIED
>
<!ELEMENT del %Flow;>
<!ATTLIST del
   %attrs;
   cite %URI; #IMPLIED
   datetime %Datetime; #IMPLIED
>
<!--================== The Anchor Element
================================-->
<!-- content is %Inline; except that anchors shouldn't be nested
-->
<!ELEMENT a %a.content;>
<!ATTLIST a
   %attrs;
   charset %Charset; #IMPLIED
   type %ContentType; #IMPLIED
   name NMTOKEN #IMPLIED
   href %URI; #IMPLIED
   hreflang %LanguageCode; #IMPLIED
   rel %LinkTypes; #IMPLIED
   rev %LinkTypes; #IMPLIED
   accesskey %Character; #IMPLIED
   shape %Shape; "rect"
   coords %Coords; #IMPLIED
   tabindex %Number; #IMPLIED
   onfocus %Script; #IMPLIED
   onblur %Script; #IMPLIED
   target %FrameTarget; #IMPLIED
>
<!--===================== Inline Elements
================================-->
<!ELEMENT span %Inline;>
<!-- generic language/style container -->
<!ATTLIST span
```

```
   %attrs;
>
<!ELEMENT bdo %Inline;>
<!-- I18N BiDi over-ride -->
<!ATTLIST bdo
   %coreattrs;
   %events;
   lang %LanguageCode; #IMPLIED
   xml:lang %LanguageCode; #IMPLIED
   dir (ltr | rtl) #REQUIRED
>
<!ELEMENT br EMPTY>
<!-- forced line break -->
<!ATTLIST br
   %coreattrs;
   clear (left | all | right | none) "none"
>
<!ELEMENT em %Inline;>
<!-- emphasis -->
<!ATTLIST em
   %attrs;
>
<!ELEMENT strong %Inline;>
<!-- strong emphasis -->
<!ATTLIST strong
   %attrs;
>
<!ELEMENT dfn %Inline;>
<!-- definitional -->
<!ATTLIST dfn
   %attrs;
>
<!ELEMENT code %Inline;>
<!-- program code -->
<!ATTLIST code
   %attrs;
>
<!ELEMENT samp %Inline;>
<!-- sample -->
<!ATTLIST samp
   %attrs;
>
<!ELEMENT kbd %Inline;>
<!-- something user would type -->
<!ATTLIST kbd
   %attrs;
>
<!ELEMENT var %Inline;>
<!-- variable -->
<!ATTLIST var
   %attrs;
>
<!ELEMENT cite %Inline;>
<!-- citation -->
<!ATTLIST cite
```

```
   %attrs;
>
<!ELEMENT abbr %Inline;>
<!-- abbreviation -->
<!ATTLIST abbr
   %attrs;
>
<!ELEMENT acronym %Inline;>
<!-- acronym -->
<!ATTLIST acronym
   %attrs;
>
<!ELEMENT q %Inline;>
<!-- inlined quote -->
<!ATTLIST q
   %attrs;
   cite %URI; #IMPLIED
>
<!ELEMENT sub %Inline;>
<!-- subscript -->
<!ATTLIST sub
   %attrs;
>
<!ELEMENT sup %Inline;>
<!-- superscript -->
<!ATTLIST sup
   %attrs;
>
<!ELEMENT tt %Inline;>
<!-- fixed pitch font -->
<!ATTLIST tt
   %attrs;
>
<!ELEMENT i %Inline;>
<!-- italic font -->
<!ATTLIST i
   %attrs;
>
<!ELEMENT b %Inline;>
<!-- bold font -->
<!ATTLIST b
   %attrs;
>
<!ELEMENT big %Inline;>
<!-- bigger font -->
<!ATTLIST big
   %attrs;
>
<!ELEMENT small %Inline;>
<!-- smaller font -->
<!ATTLIST small
   %attrs;
>
<!ELEMENT u %Inline;>
<!-- underline -->
```

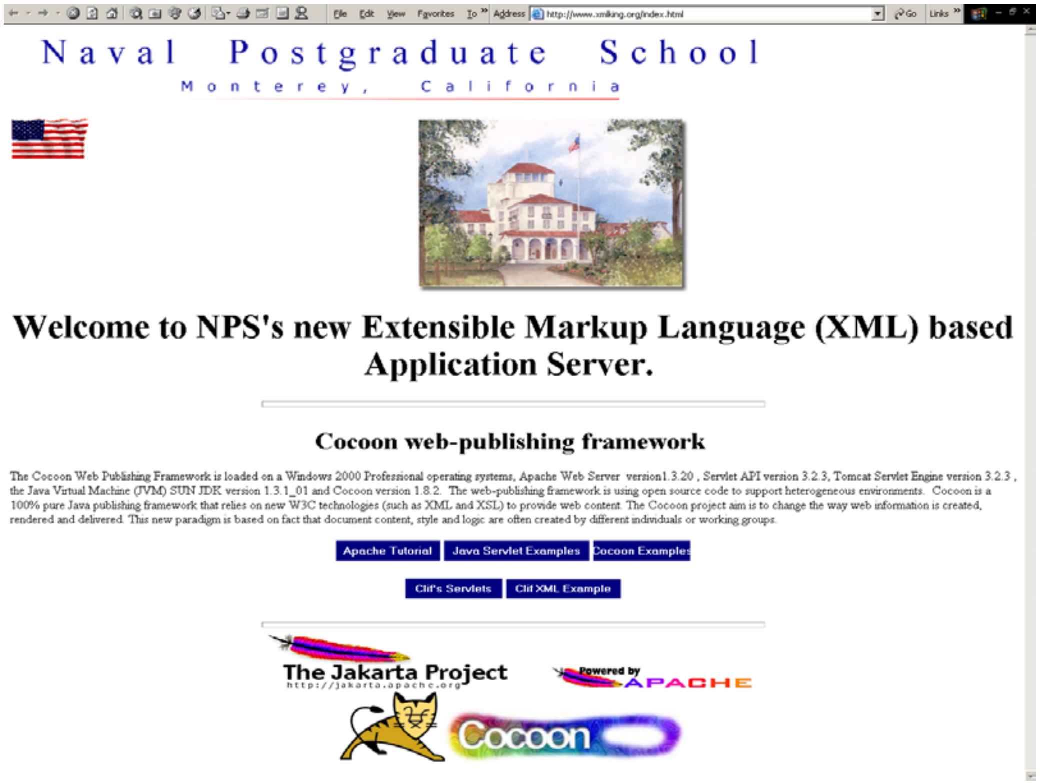
```
<!ATTLIST u
   %attrs;
>
<!ELEMENT s %Inline;>
<!-- strike-through -->
<!ATTLIST s
   %attrs;
>
<!ELEMENT strike %Inline;>
<!-- strike-through -->
<!ATTLIST strike
   %attrs;
>
<!ELEMENT basefont EMPTY>
<!-- base font size -->
<!ATTLIST basefont
   id ID #IMPLIED
   size CDATA #REQUIRED
   color %Color; #IMPLIED
   face CDATA #IMPLIED
>
<!ELEMENT font %Inline;>
<!-- local change to font -->
<!ATTLIST font
   %coreattrs;
   %i18n;
   size CDATA #IMPLIED
   color %Color; #IMPLIED
   face CDATA #IMPLIED
>
<!--====================                                    Object
======================================-->
<!--
  object is used to embed objects as part of HTML pages.
  param elements should precede other content. Parameters
  can also be expressed as attribute/value pairs on the
  object element itself when brevity is desired.
-->
<!ELEMENT object (#PCDATA | param | %block; | form | %inline; |
%misc;)*>
<!ATTLIST object
   %attrs;
   declare (declare) #IMPLIED
   classid %URI; #IMPLIED
   codebase %URI; #IMPLIED
   data %URI; #IMPLIED
   type %ContentType; #IMPLIED
   codetype %ContentType; #IMPLIED
   archive %UriList; #IMPLIED
   standby %Text; #IMPLIED
   height %Length; #IMPLIED
   width %Length; #IMPLIED
   usemap %URI; #IMPLIED
   name NMTOKEN #IMPLIED
   tabindex %Number; #IMPLIED
   align %ImgAlign; #IMPLIED
   border %Pixels; #IMPLIED
   hspace %Pixels; #IMPLIED
   vspace %Pixels; #IMPLIED
>
<!--
  param is used to supply a named property value.
  In XML it would seem natural to follow RDF and support an
  abbreviated syntax where the param elements are replaced
  by attribute value pairs on the object start tag.
-->
<!ELEMENT param EMPTY>
<!ATTLIST param
   id ID #IMPLIED
   name CDATA #REQUIRED
   value CDATA #IMPLIED
   valuetype (data | ref | object) "data"
   type %ContentType; #IMPLIED
>
<!--===================                Java                applet
==================================-->
<!--
  One of code or object attributes must be present.
  Place param elements before other content.
-->
<!ELEMENT applet (#PCDATA | param | %block; | form | %inline; |
%misc;)*>
<!ATTLIST applet
   %coreattrs;
   codebase %URI; #IMPLIED
   archive CDATA #IMPLIED
   code CDATA #IMPLIED
   object CDATA #IMPLIED
   alt %Text; #IMPLIED
   name NMTOKEN #IMPLIED
   width %Length; #REQUIRED
   height %Length; #REQUIRED
   align %ImgAlign; #IMPLIED
   hspace %Pixels; #IMPLIED
   vspace %Pixels; #IMPLIED
>
<!--===================                                   Images
===========================================-->
<!--
   To avoid accessibility problems for people who aren't
   able to see the image, you should provide a text
   description using the alt and longdesc attributes.
   In addition, avoid the use of server-side image maps.
-->
<!ELEMENT img EMPTY>
<!ATTLIST img
   %attrs;
   src %URI; #REQUIRED
   alt %Text; #REQUIRED
   name NMTOKEN #IMPLIED
   longdesc %URI; #IMPLIED
   height %Length; #IMPLIED
   width %Length; #IMPLIED
   usemap %URI; #IMPLIED
   ismap (ismap) #IMPLIED
   align %ImgAlign; #IMPLIED
   border %Length; #IMPLIED
   hspace %Pixels; #IMPLIED
   vspace %Pixels; #IMPLIED
>
<!-- usemap points to a map element which may be in this document
  or  an  external  document,  although  the  latter  is  not  widely
supported -->
<!--==================        Client-side        image        maps
============================-->
<!-- These can be placed in the same document or grouped in a
     separate document although this isn't yet widely supported -
->
<!ELEMENT map ((%block; | form | %misc;)+ | area+)>
<!ATTLIST map
   %i18n;
   %events;
   id ID #REQUIRED
   class CDATA #IMPLIED
   style %StyleSheet; #IMPLIED
   title %Text; #IMPLIED
   name CDATA #IMPLIED
>
<!ELEMENT area EMPTY>
<!ATTLIST area
   %attrs;
   shape %Shape; "rect"
   coords %Coords; #IMPLIED
   href %URI; #IMPLIED
   nohref (nohref) #IMPLIED
   alt %Text; #REQUIRED
   tabindex %Number; #IMPLIED
   accesskey %Character; #IMPLIED
   onfocus %Script; #IMPLIED
   onblur %Script; #IMPLIED
   target %FrameTarget; #IMPLIED
>
<!--================                                         Forms
===============================================-->
<!ELEMENT form %form.content;>
<!-- forms shouldn't be nested -->
<!ATTLIST form
   %attrs;
   action %URI; #REQUIRED
   method (get | post) "get"
   name NMTOKEN #IMPLIED
   enctype %ContentType; "application/x-www-form-urlencoded"
   onsubmit %Script; #IMPLIED
   onreset %Script; #IMPLIED
   accept %ContentTypes; #IMPLIED
   accept-charset %Charsets; #IMPLIED
   target %FrameTarget; #IMPLIED
>
<!--
  Each label must not contain more than ONE field
  Label elements shouldn't be nested.
-->
<!ELEMENT label %Inline;>
<!ATTLIST label
   %attrs;
   for IDREF #IMPLIED
   accesskey %Character; #IMPLIED
   onfocus %Script; #IMPLIED
   onblur %Script; #IMPLIED
>
<!ENTITY % InputType "(text | password | checkbox |
    radio | submit | reset |
    file | hidden | image | button)">
<!-- the name attribute is required for all but submit & reset --
>
<!ELEMENT input EMPTY>
<!-- form control -->
<!ATTLIST input
   %attrs;
   type %InputType; "text"
   name CDATA #IMPLIED
   value CDATA #IMPLIED
   checked (checked) #IMPLIED
   disabled (disabled) #IMPLIED
   readonly (readonly) #IMPLIED
   size CDATA #IMPLIED
   maxlength %Number; #IMPLIED
   src %URI; #IMPLIED
   alt CDATA #IMPLIED
   usemap %URI; #IMPLIED
   tabindex %Number; #IMPLIED
   accesskey %Character; #IMPLIED
   onfocus %Script; #IMPLIED
   onblur %Script; #IMPLIED
   onselect %Script; #IMPLIED
   onchange %Script; #IMPLIED
   accept %ContentTypes; #IMPLIED
   align %ImgAlign; #IMPLIED
>
<!ELEMENT select (optgroup | option)+>
<!-- option selector -->
<!ATTLIST select
   %attrs;
   name CDATA #IMPLIED
   size %Number; #IMPLIED
   multiple (multiple) #IMPLIED
   disabled (disabled) #IMPLIED
   tabindex %Number; #IMPLIED
   onfocus %Script; #IMPLIED
   onblur %Script; #IMPLIED
   onchange %Script; #IMPLIED
>
<!ELEMENT optgroup (option)+>
<!-- option group -->
<!ATTLIST optgroup
   %attrs;
   disabled (disabled) #IMPLIED
   label %Text; #REQUIRED
>
<!ELEMENT option (#PCDATA)>
<!-- selectable choice -->
<!ATTLIST option
   %attrs;
   selected (selected) #IMPLIED
   disabled (disabled) #IMPLIED
   label %Text; #IMPLIED
   value CDATA #IMPLIED
>
<!ELEMENT textarea (#PCDATA)>
<!-- multi-line text field -->
<!ATTLIST textarea
   %attrs;
   name CDATA #IMPLIED
   rows %Number; #REQUIRED
   cols %Number; #REQUIRED
   disabled (disabled) #IMPLIED
   readonly (readonly) #IMPLIED
   tabindex %Number; #IMPLIED
   accesskey %Character; #IMPLIED
   onfocus %Script; #IMPLIED
   onblur %Script; #IMPLIED
   onselect %Script; #IMPLIED
   onchange %Script; #IMPLIED
>
<!--
  The fieldset element is used to group form fields.
  Only one legend element should occur in the content
  and if present should only be preceded by whitespace.
-->
<!ELEMENT fieldset (#PCDATA | legend | %block; | form | %inline;
| %misc;)*>
<!ATTLIST fieldset
   %attrs;
>
<!ENTITY % LAlign "(top|bottom|left|right)">
<!ELEMENT legend %Inline;>
<!-- fieldset label -->
<!ATTLIST legend
   %attrs;
   accesskey %Character; #IMPLIED
   align %LAlign; #IMPLIED
>
<!--
 Content is %Flow; excluding a, form, form controls, iframe
-->
```

```
<!ELEMENT button %button.content;>
<!-- push button -->
<!ATTLIST button
   %attrs;
   name CDATA #IMPLIED
   value CDATA #IMPLIED
   type (button | submit | reset) "submit"
   disabled (disabled) #IMPLIED
   tabindex %Number; #IMPLIED
   accesskey %Character; #IMPLIED
   onfocus %Script; #IMPLIED
   onblur %Script; #IMPLIED
>
<!-- single-line text input control (DEPRECATED) -->
<!ELEMENT isindex EMPTY>
<!ATTLIST isindex
   %coreattrs;
   %i18n;
   prompt %Text; #IMPLIED
>
<!--=======================                            Tables
=========================================-->
<!-- Derived from IETF HTML table standard, see [RFC1942] -->
<!--
 The border attribute sets the thickness of the frame around the
 table. The default units are screen pixels.

 The frame attribute specifies which parts of the frame around
 the table should be rendered. The values are not the same as
 CALS to avoid a name clash with the valign attribute.
-->
<!ENTITY                          %                          TFrame
"(void|above|below|hsides|lhs|rhs|vsides|box|border)">
<!--
 The rules attribute defines which rules to draw between cells:

 If rules is absent then assume:
     "none" if border is absent or border="0" otherwise "all"
-->
<!ENTITY % TRules "(none | groups | rows | cols | all)">
<!-- horizontal placement of table relative to document -->
<!ENTITY % TAlign "(left|center|right)">
<!-- horizontal alignment attributes for cell contents

  char        alignment char, e.g. char=':'
  charoff     offset for alignment char
-->
<!ENTITY             %             cellhalign             "align
(left|center|right|justify|char) #IMPLIED
   char       %Character;    #IMPLIED
   charoff    %Length;       #IMPLIED">
<!-- vertical alignment attributes for cell contents -->
<!ENTITY % cellvalign "valign       (top|middle|bottom|baseline)
#IMPLIED">
<!ELEMENT table (caption?, (col* | colgroup*), thead?, tfoot?,
(tbody+ | tr+))>
<!ELEMENT caption %Inline;>
<!ELEMENT thead (tr)+>
<!ELEMENT tfoot (tr)+>
<!ELEMENT tbody (tr)+>
<!ELEMENT colgroup (col)*>
<!ELEMENT col EMPTY>
<!ELEMENT tr (th | td)+>
<!ELEMENT th %Flow;>
<!ELEMENT td %Flow;>
<!ATTLIST table
  %attrs;
  summary %Text; #IMPLIED
  width %Length; #IMPLIED
  border %Pixels; #IMPLIED
  frame %TFrame; #IMPLIED
  rules %TRules; #IMPLIED
  cellspacing %Length; #IMPLIED
  cellpadding %Length; #IMPLIED
  align %TAlign; #IMPLIED
  bgcolor %Color; #IMPLIED
>
<!ENTITY % CAlign "(top|bottom|left|right)">
<!ATTLIST caption
  %attrs;
  align %CAlign; #IMPLIED
>
<!--
colgroup groups a set of col elements. It allows you to group
several semantically related columns together.
-->
<!ATTLIST colgroup
  %attrs;
  span %Number; "1"
  width %MultiLength; #IMPLIED
  %cellhalign;
  %cellvalign;
>
<!--
 col elements define the alignment properties for cells in
 one or more columns.

 The width attribute specifies the width of the columns, e.g.

     width=64        width in screen pixels
     width=0.5*      relative width of 0.5

 The span attribute causes the attributes of one
 col element to apply to more than one column.
-->
<!ATTLIST col
  %attrs;
  span %Number; "1"
  width %MultiLength; #IMPLIED
  %cellhalign;
```

```
   %cellvalign;
>
<!--
    Use thead to duplicate headers when breaking table
    across page boundaries, or for static headers when
    tbody sections are rendered in scrolling panel.

    Use tfoot to duplicate footers when breaking table
    across page boundaries, or for static footers when
    tbody sections are rendered in scrolling panel.

    Use multiple tbody sections when rules are needed
    between groups of table rows.
-->
<!ATTLIST thead
   %attrs;
   %cellhalign;
   %cellvalign;
>
<!ATTLIST tfoot
   %attrs;
   %cellhalign;
   %cellvalign;
>
<!ATTLIST tbody
   %attrs;
   %cellhalign;
   %cellvalign;
>
<!ATTLIST tr
   %attrs;
   %cellhalign;
   %cellvalign;
   bgcolor %Color; #IMPLIED
>
<!-- Scope is simpler than headers attribute for common tables --
>
<!ENTITY % Scope "(row|col|rowgroup|colgroup)">
<!-- th is for headers, td for data and for cells acting as both
-->
<!ATTLIST th
   %attrs;
   abbr %Text; #IMPLIED
   axis CDATA #IMPLIED
   headers IDREFS #IMPLIED
   scope %Scope; #IMPLIED
   rowspan %Number; "1"
   colspan %Number; "1"
   %cellhalign;
   %cellvalign;
   nowrap (nowrap) #IMPLIED
   bgcolor %Color; #IMPLIED
   width %Pixels; #IMPLIED
   height %Pixels; #IMPLIED
>
```

```
<!ATTLIST td
   %attrs;
   abbr %Text; #IMPLIED
   axis CDATA #IMPLIED
   headers IDREFS #IMPLIED
   scope %Scope; #IMPLIED
   rowspan %Number; "1"
   colspan %Number; "1"
   %cellhalign;
   %cellvali; #IMPLIED
   width %Pixels; gn;
   nowrap (nowrap) #IMPLIED
   bgcolor %Color#IMPLIED
   height %Pixels; #IMPLIED
>
```

F. XHTML BOWSER OUTPUT


File Edit View Favorites Tools
Address http://www.xmlking.org/index.html
Go
Links

Naval Postgraduate School

Monterey, California

Welcome to NPS's new Extensible Markup Language (XML) based Application Server.

Cocoon web-publishing framework

The Cocoon Web Publishing Framework is loaded on a Windows 2000 Professional operating systems, Apache Web Server version1.3.20 , Servlet API version 3.2.3, Tomcat Servlet Engine version 3.2.3 , the Java Virtual Machine (JVM) SUN JDK version 1.3.1_01 and Cocoon version 1.8.2. The web-publishing framework is using open source code to support heterogeneous environments. Cocoon is a 100% pure Java publishing framework that relies on new W3C technologies (such as XML and XSL) to provide web content. The Cocoon project aim is to change the way web information is created, rendered and delivered. This new paradigm is based on fact that document content, style and logic are often created by different individuals or working groups.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D. JAVA SERVLET AND JSP EXAMPLES

A . INTRODUCTION

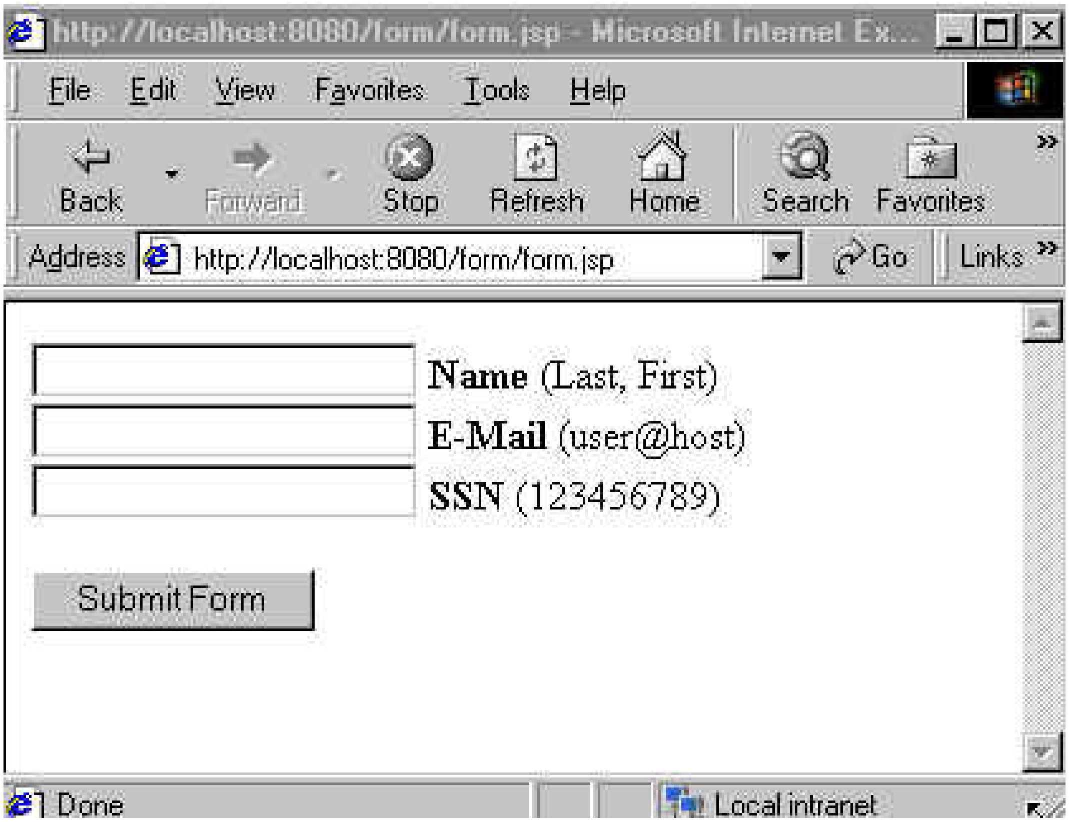
This JSP The JSP Form Page and JSP Handle Form [Tom01]. The JSP form page will be the start page and the error page in this example. Initially it will simply display an HTML form (Appendix D Section C). It will target a form validation Servlet (Appendix D Section D). If the Servlet detects errors in the user input, it will forward the request back to the JSP page(Appendix D Section B). The JSP page will notify the user of errors and redisplay the form with the original user inputs intact.

B. JSP FORM

```
<jsp:useBean id="form" class="FormBean">
  <jsp:setProperty name="form" property="*"/>
</jsp:useBean>
<html>
<body bgcolor="white">
<%
  String[] errors = (String[])request.getAttribute("errors");
  if (errors != null && errors.length > 0) {
%>
<b>Please Correct the Following Errors</b>
<ul>
<% for (int i=0; i < errors.length; i++) { %>
<li> <%= errors[i] %>
<% } %>
</ul>
<% } %>
```

```
<form action="<%= request.getContextPath()
%>/Servlet/HandleForm" method="post">
<input type="text" name="name"
value="<jsp:getProperty name="form" property="name"/>">
<b>Name</b> (Last, First)<br>
<input type="text" name="email"
value="<jsp:getProperty name="form" property="email"/>">
<b>E-Mail</b> (user@host)<br>
<input type="text" name="ssn"
value="<jsp:getProperty name="form" property="ssn"/>">
<b>SSN</b> (123456789)<br>
<p>
<input type="submit" value="Submit Form">
</form>
</body>
</html>
```

C. HTML FORM



Name (Last, First)

E-Mail (user@host)

SSN (123456789)

D. SERVLET

```
import java.io.*;
import javax.Servlet.*;
import javax.Servlet.http.*;
import java.util.*;
public class HandleForm extends HttpServlet {
   public       void       service(HttpServletRequest       req,
HttpServletResponse res)
                              throws            ServletException,
IOException {
   Vectorimport java.io.*;
import javax.Servlet.*;
import javax.Servlet.http.*;
import java.util.*;
public class HandleForm extends HttpServlet {
   public       void       service(HttpServletRequest       req,
HttpServletResponse res)
                              throws            ServletException,
IOException {
   Vector errors = new Vector();
   String name = req.getParameter("name");
   String ssn = req.getParameter("ssn");
   String email = req.getParameter("email");
   if (! isValidName(name))
      errors.add("Please specify the name as Last, First");
   if (! isValidEmail(email))
      errors.add("Email address must contain an @ symbol");
   if (! isValidSSN(ssn))
      errors.add("Please   specify   a   valid   SSN   number,   no
dashes");
   String next;
   if (errors.size() == 0) {      // data is OK
      next = "thanks.jsp";
   }    else {      // data has errors, try again
      String[]    errorArray    =    (String[])errors.toArray(new
String[0]);
      req.setAttribute("errors", errorArray);
      next = "form.jsp";
   }
   RequestDispatcher rd;
   rd = getServletContext().getRequestDispatcher("/" + next);
   rd.forward(req, res);
   }
   private boolean isValidSSN(String ssn) {      // check for 9
characters, no dashes
      return (ssn.length() == 9 && ssn.indexOf("-") == -1);
   }
   private boolean isValidEmail(String email) {      // check an
"@" somewhere after the 1st character
      return (email.indexOf("@") > 0);
   }
   private boolean isValidName(String name) {      // should be
Last, First - check for the comma
      return (name.indexOf(",") != -1);
   }
}
 errors = new Vector();
   String name = req.getParameter("name");
   String ssn = req.getParameter("ssn");
   String email = req.getParameter("email");
   if (! isValidName(name))
      errors.add("Please specify the name as Last, First");
   if (! isValidEmail(email))
      errors.add("Email address must contain an @ symbol");
   if (! isValidSSN(ssn))
      errors.add("Please   specify   a   valid   SSN   number,   no
dashes");
   String next;
   if (errors.size() == 0) {      // data is OK
```

```
      next = "thanks.jsp";
   }    else {      // data has errors, try again
      String[]    errorArray    =    (String[])errors.toArray(new
String[0]);
      req.setAttribute("errors", errorArray);
      next = "form.jsp";
   }
   RequestDispatcher rd;
   rd = getServletContext().getRequestDispatcher("/" + next);
   rd.forward(req, res);
   }
   private boolean isValidSSN(String ssn) {      // check for 9
characters, no dashes
      return (ssn.length() == 9 && ssn.indexOf("-") == -1);
   }
   private boolean isValidEmail(String email) {      // check an
"@" somewhere after the 1st character
      return (email.indexOf("@") > 0);
   }
   private boolean isValidName(String name) {      // should be
Last, First - check for the comma
      return (name.indexOf(",") != -1);
   }
}
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX E. WEB PUBLISHING FRAMEWORK'S SOURCE CODE USING SERVLETS FOR GENERIC DATABASE ACCESS

A . INTRODUCTION

The Web Publishing Framework prototype site uses XML and Java technologies in the Web site is design. Section B, C and D are Java Servlet Source Code GenericXMLJdbcAccess.Class, JdbcXmlDB.Class, and JdbcXhtml.Class. Section E is a XML over HTTP document output. Section F is the JDBCHTML.xsl stylesheet source code. Section G XHTML interface form and XHTML source code

B. GENERICXMLJDBCACCESS JAVA SOURCE CODE

```
Package jdbc.xml;

import java.sql.*;
import java.util.*;

/**
 * Title:        GenericXMLJdbcAccess CLASS
 * Description:  The Class encapsulates the functionality
 *               required to query a JDBC data source and return
 *               the results that query as a well-formed xml
 *               Document.
 * Copyright:    Copyright (c) 2001
 * Company: www.xmlking.org
 * @author Clifton Williams
 * @version 1.0
 */

public class GenericXMLJdbcAccess {



  public GenericXMLJdbcAccess () {
     super();
  }

/**
 * This method apply XML encoding rules
 *
 */
 String encodeXML(String sData)
 {
```

```
 String[] before = {"&","<",">","\"","\'"};
 String[] after =  {"&","<",">",""","&apos;"};
 if (sData!=null)
 {
    for(int i=0; i < before.length; i++)
    {
       sData = Replace(sData, before[i], after[i]);
    }
 }
 else {sData="";}
 return sData;
 }


 /**
  * Used to replace code frament
  */

 String Replace (String content, String oldWord, String newWord)
 {
    int position = content.indexOf(oldWord);
    while (position > -1)
    {
       content =content.substring(0,position) + newWord +
          content.substring(position + oldWord.length());
          position = content.indexOf(oldWord,position
+newWord.length());
    }
    return content;
   }

/**
 * This method specified JDBC result set as an XML document
 *
 */

 String writeXML(ResultSet rs)
 {
    StringBuffer strResults = new StringBuffer
       ("<?xml version=\"1.0\" encoding=\"ISO-8859-
1\"?>\r\n<resultset>\r\n");
    try
    {
       ResultSetMetaData rsMetadata =rs.getMetaData();
       int intFields = rsMetadata.getColumnCount();
       strResults.append("<metadata>\r\n");
       for(int h=1;h<= intFields; h++)
       {
          strResults.append("<field name=\" " +
rsMetadata.getColumnName(h) +
            "\" datatype=\"" + rsMetadata.getColumnTypeName(h) +
"\"/>\r\n");
       }
       strResults.append("</metadata>\r\n<records>\r\n");
       while (rs.next())
       {
          strResults.append("<record>\r\n");
          for(int i=1; i<=intFields; i++)
          {
             strResults.append("<field name=\" " +
rsMetadata.getColumnName(i)
             + "\">" + encodeXML(rs.getString(i)) +
"</field>\r\n");
          }
          strResults.append("</record>\r\n");
       }
    }catch(Exception e) {}
    strResults.append("</records>\r\n</resultset>");
    return strResults.toString();
    }

/**
 *
 *  Establish a connection to a database and executes the SQL
 *  Query returns String Resultset formated in XML serialized
 *  stream
 *
 */

public String execute(String driver, String url, String uid,
String pwd, String sql)
 {
 String output = new String();
 try
 {
    Class.forName(driver);
    Connection conn = DriverManager.getConnection(url,uid,pwd);
    Statement s = conn.createStatement();
    ResultSet rs= s.executeQuery(sql);
    output = writeXML(rs);
    rs.close();
    conn.close();
 }
 catch(Exception e)
 {
    output = "<error>" + encodeXML(e.toString()) + "</error>";
 }
 return output;
 }
}
```

C. JDBCXMLDB JAVA SOURCE CODE

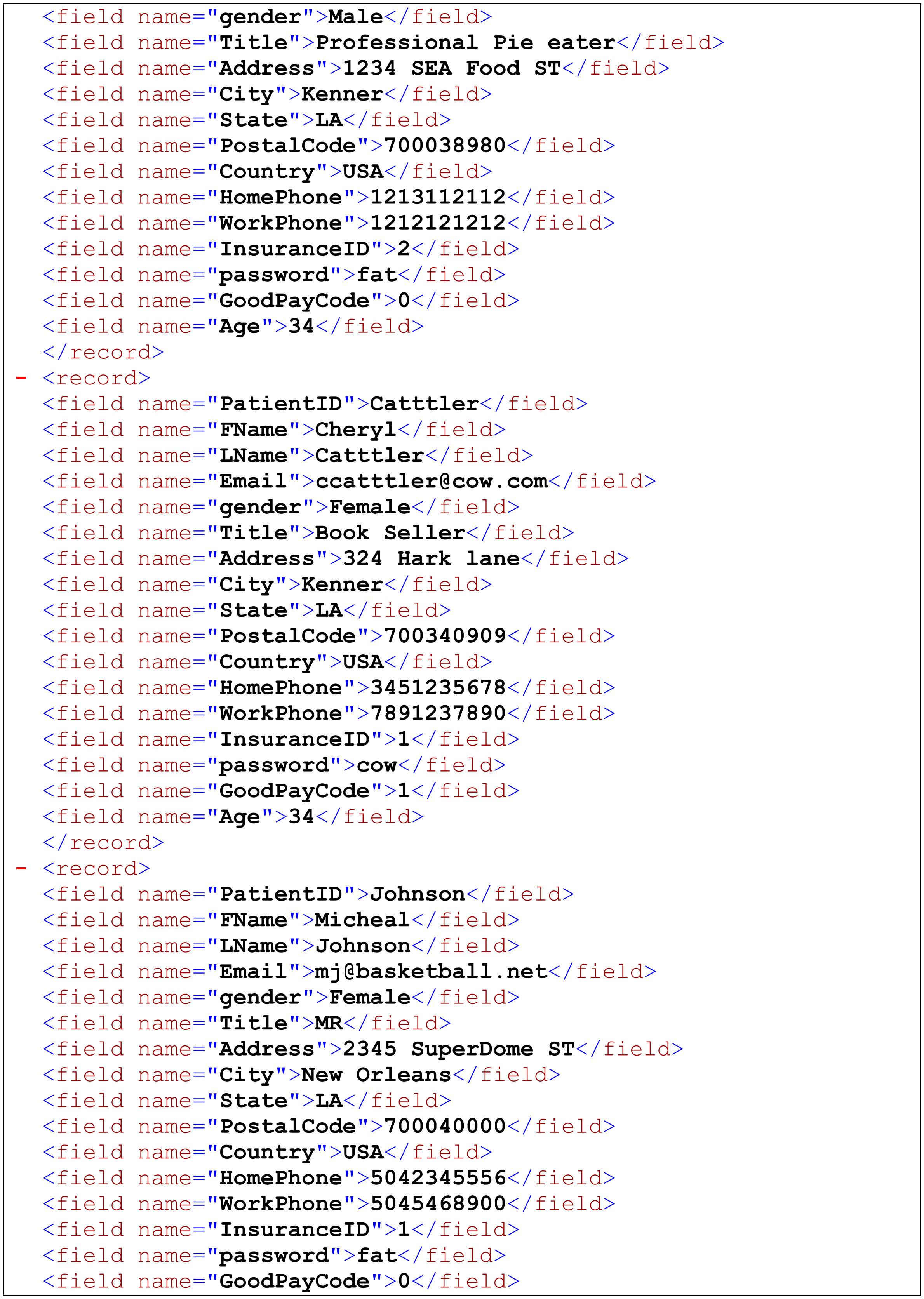
```
package jdbc.xml;

import javax.Servlet.*;
import javax.Servlet.http.*;
import java.io.*;

/**
 * Title:        JdbcXmlDB CLASS
 * Description:  The Class encapsulates the functionality
 *               required  to query a JDBC data source and return
 *               the results that query as a well-formed
 *               xml Document.
 * Copyright:    Copyright (c) 2001
 * Company:      www.xmlking.org
 * @author       Clifton Williams
 * @version 1.1
 */


public class JdbcXmlDB extends HttpServlet
{

  public void doGet(HttpServletRequest request,
HttpServletResponse response)
     throws IOException, ServletException
  {

  response.setContentType("text/xml");
  PrintWriter out = response.getWriter();
  GenericXMLJdbcAccess searchOBJ = new GenericXMLJdbcAccess ();
  out.println(searchOBJ.execute(request.getParameter("driver"),
     request.getParameter("jdbcurl"),
request.getParameter("uid"),
     request.getParameter("pwd"),request.getParameter("sql")));

  }

public void doPost(HttpServletRequest request,
HttpServletResponse response)
     throws IOException, ServletException

  {

  doGet(request,response);

  }

}
```

D. JDBCXHTMLDB JAVA SOURCE CODE

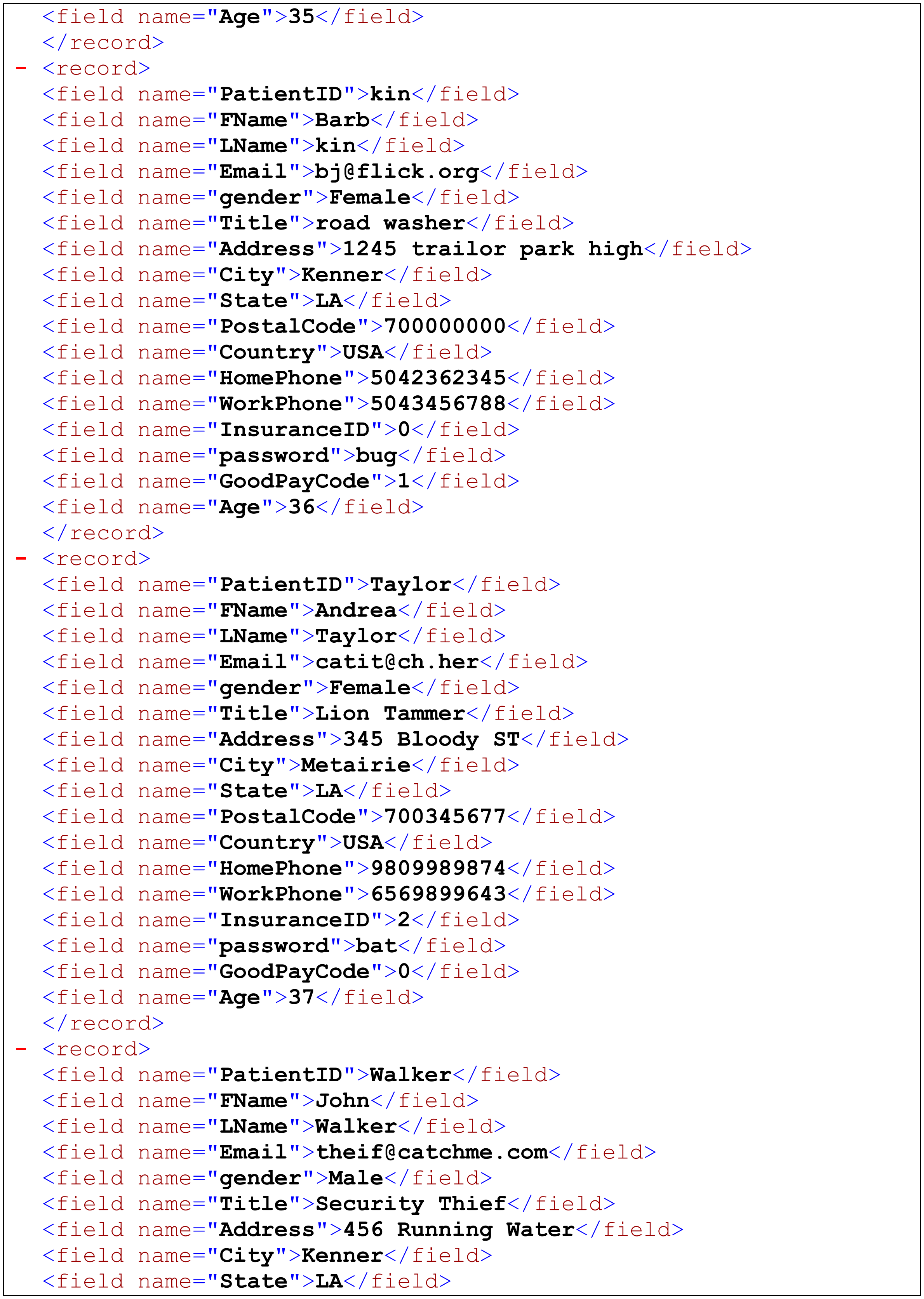
```
Package jdbc.xml;

import javax.Servlet.*;
import javax.Servlet.http.*;
import java.io.*;
import org.xml.sax.SAXException;
import org.apache.xalan.xslt.*;


/**
 * Title:          JdbcXhtmlDB CLASS
 * Description:    The Class encapsulates the functionality required to
 *                 query a JDBC data source and return the results that
 *                 query as a well-formed xml Document. Then applies a
 *                 stylesheet to convert the output to HTML
 * Copyright:      Copyright (c) 2001
 * Company:        www.xmlking.org
 * @author         Clifton Williams
 * @version 1.1
 */

public class JdbcXhtmlDB extends HttpServlet
{

  public void doGet(HttpServletRequest req, HttpServletResponse res)
     throws IOException, ServletException
  {
  String qryPath;
  String xsltSystemId = new File("C:/jakarta-tomcat-
3.2.3/Webapps/jdbcxml/WEB-
INF/classes/jdbc/xml/JDBCHTML.xsl").toURL().toExternalForm();


  if(req.getPathInfo()==null)
  {
     qryPath =
getServletConfig().getServletContext().getRealPath(req.getServletPath());

  }
  else
  {
  qryPath = req.getPathTranslated();
  }
  res.setContentType("text/html");
  PrintWriter xmlout = res.getWriter();
  GenericXMLJdbcAccess searchOBJ = new GenericXMLJdbcAccess ();
  String output = searchOBJ.execute(req.getParameter("driver"),
     req.getParameter("jdbcurl"), req.getParameter("uid"),
     req.getParameter("pwd"),req.getParameter("sql"));
  try
  {
  XSLTProcessor proc = XSLTProcessorFactory.getProcessor();
  proc.process(new XSLTInputSource(new java.io.StringReader(output)),
     new XSLTInputSource(xsltSystemId), new XSLTResultTarget(xmlout));
  }
  catch(SAXException se)
  {
     throw new ServletException(se);
  }
 }

  public void doPost(HttpServletRequest req, HttpServletResponse res)
     throws IOException, ServletException
  {
  doGet(req,res);
  }
}
```

E. XML DATABASE XML OVER HTTP OUTPUT

```
  <?xml version="1.0" encoding="ISO-8859-1" ?>
- <resultset>
- <metadata>
  <field name="PatientID" datatype="VARCHAR" />
  <field name="FName" datatype="VARCHAR" />
  <field name="LName" datatype="VARCHAR" />
  <field name="Email" datatype="VARCHAR" />
  <field name="gender" datatype="VARCHAR" />
  <field name="Title" datatype="VARCHAR" />
  <field name="Address" datatype="VARCHAR" />
  <field name="City" datatype="VARCHAR" />
  <field name="State" datatype="VARCHAR" />
  <field name="PostalCode" datatype="VARCHAR" />
  <field name="Country" datatype="VARCHAR" />
  <field name="HomePhone" datatype="VARCHAR" />
  <field name="WorkPhone" datatype="VARCHAR" />
  <field name="InsuranceID" datatype="INTEGER" />
  <field name="password" datatype="VARCHAR" />
  <field name="GoodPayCode" datatype="BIT" />
  <field name="Age" datatype="VARCHAR" />
  </metadata>
- <records>
- <record>
  <field name="PatientID">Boy</field>
  <field name="FName">Fat</field>
  <field name="LName">Boy</field>
  <field name="Email">fboy@help.me</field>
```



```
  <field name="gender">Male</field>
  <field name="Title">Professional Pie eater</field>
  <field name="Address">1234 SEA Food ST</field>
  <field name="City">Kenner</field>
  <field name="State">LA</field>
  <field name="PostalCode">700038980</field>
  <field name="Country">USA</field>
  <field name="HomePhone">1213112112</field>
  <field name="WorkPhone">1212121212</field>
  <field name="InsuranceID">2</field>
  <field name="password">fat</field>
  <field name="GoodPayCode">0</field>
  <field name="Age">34</field>
  </record>
- <record>
  <field name="PatientID">Catttler</field>
  <field name="FName">Cheryl</field>
  <field name="LName">Catttler</field>
  <field name="Email">ccatttler@cow.com</field>
  <field name="gender">Female</field>
  <field name="Title">Book Seller</field>
  <field name="Address">324 Hark lane</field>
  <field name="City">Kenner</field>
  <field name="State">LA</field>
  <field name="PostalCode">700340909</field>
  <field name="Country">USA</field>
  <field name="HomePhone">3451235678</field>
  <field name="WorkPhone">7891237890</field>
  <field name="InsuranceID">1</field>
  <field name="password">cow</field>
  <field name="GoodPayCode">1</field>
  <field name="Age">34</field>
  </record>
- <record>
  <field name="PatientID">Johnson</field>
  <field name="FName">Micheal</field>
  <field name="LName">Johnson</field>
  <field name="Email">mj@basketball.net</field>
  <field name="gender">Female</field>
  <field name="Title">MR</field>
  <field name="Address">2345 SuperDome ST</field>
  <field name="City">New Orleans</field>
  <field name="State">LA</field>
  <field name="PostalCode">700040000</field>
  <field name="Country">USA</field>
  <field name="HomePhone">5042345556</field>
  <field name="WorkPhone">5045468900</field>
  <field name="InsuranceID">1</field>
  <field name="password">fat</field>
  <field name="GoodPayCode">0</field>
```



```
  <field name="Age">35</field>
  </record>
- <record>
  <field name="PatientID">kin</field>
  <field name="FName">Barb</field>
  <field name="LName">kin</field>
  <field name="Email">bj@flick.org</field>
  <field name="gender">Female</field>
  <field name="Title">road washer</field>
  <field name="Address">1245 trailor park high</field>
  <field name="City">Kenner</field>
  <field name="State">LA</field>
  <field name="PostalCode">700000000</field>
  <field name="Country">USA</field>
  <field name="HomePhone">5042362345</field>
  <field name="WorkPhone">5043456788</field>
  <field name="InsuranceID">0</field>
  <field name="password">bug</field>
  <field name="GoodPayCode">1</field>
  <field name="Age">36</field>
  </record>
- <record>
  <field name="PatientID">Taylor</field>
  <field name="FName">Andrea</field>
  <field name="LName">Taylor</field>
  <field name="Email">catit@ch.her</field>
  <field name="gender">Female</field>
  <field name="Title">Lion Tammer</field>
  <field name="Address">345 Bloody ST</field>
  <field name="City">Metairie</field>
  <field name="State">LA</field>
  <field name="PostalCode">700345677</field>
  <field name="Country">USA</field>
  <field name="HomePhone">9809989874</field>
  <field name="WorkPhone">6569899643</field>
  <field name="InsuranceID">2</field>
  <field name="password">bat</field>
  <field name="GoodPayCode">0</field>
  <field name="Age">37</field>
  </record>
- <record>
  <field name="PatientID">Walker</field>
  <field name="FName">John</field>
  <field name="LName">Walker</field>
  <field name="Email">theif@catchme.com</field>
  <field name="gender">Male</field>
  <field name="Title">Security Thief</field>
  <field name="Address">456 Running Water</field>
  <field name="City">Kenner</field>
  <field name="State">LA</field>
  <field name="PostalCode">703401233</field>
  <field name="Country">USA</field>
  <field name="HomePhone">4567867211</field>
  <field name="WorkPhone">6575676567</field>
  <field name="InsuranceID">3</field>
  <field name="password">hog</field>
  <field name="GoodPayCode">1</field>
  <field name="Age">38</field>
  </record>
  </records>
  </resultset>
```

F. JDBCHTML STYLESHEET XML SOURCE CODE

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- developed with XML Spy v4.0 U (http://www.xmlspy.com) by Clifton Williams (Naval
Postgraduate School) -->
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
<xsl:template match="/">
<xsl:apply-templates/>
</xsl:template>
<xsl:template match="resultset">
<html>
<head>
<title>
This is an Database to XML Servlet in HTML
</title>
</head>
<body>
<img src="http:\\www.xmlking.org\images\HerrmannHallHomePage.jpg" alt="Herman
Hall"/>
<h1 align="center">
Clif's XML to HTML Database example
</h1>
<table border="1" cellspacing="0" cellpadding="5" align="center">
  <xsl:apply-templates select="metadata"/>
  <xsl:apply-templates select="//resultset/records"/>
</table>
</body>
</html>
</xsl:template>
<xsl:template match="metadata">
 <tr bgcolor="#FFD700">
 <xsl:apply-templates/>
 </tr>
</xsl:template>
```

```
<xsl:template match="metadata/field">
  <td><b><xsl:value-of select="@name"/></b></td>
  <xsl:apply-templates/>
  </xsl:template>
<xsl:template match="records">
<xsl:apply-templates/>
</xsl:template>
<xsl:template match="record">
<tr>
 <xsl:for-each select="field">
 <td><xsl:value-of select="."/>
 </td>
 </xsl:for-each>
</tr>
</xsl:template>
<xsl:template match="error">
<html>
<head>
<title>
This is an Database to XML Servlet in HTML
</title>
</head>
<body>
This request has the following error:<xsl:value-of select="."/>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

G. XHTML INTERFACE FORM XHTML SOURCE CODE

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<!-- developed with XML Spy v4.0 U (http://www.xmlspy.com) by Clifton Williams (Naval
Postgraduate School) -->

<head>

<title>NPS XML Server Home Page</title>

</head>

<body bgcolor="#FFFFFF" text="#000000" link="#0000FF" vlink=
"#000080" alink="#FF0000">
<p><img border="0" src="images/NPS_r01_c1.gif" width="969"
height="90" align="middle" /></p>


<p align="left"><img border="0" src="images/Flag.gif" align=
"left" width="100" height="55" /></p>


<p align="center"><img border="0" src=
"images/HerrmannHallHomePage.jpg" width="335" height="232" align=
"middle" /></p>

<hr width="50%" size="8" />
<h1 align="center">Cocoon Web-publishing framework </h1>

<h2 align="center">Database Connection and Query page</h2>

<hr />

<form action="/jdbcxml/Servlet/jdbc.xml.JDBCHTML" method="POST">
<table border="0">

<tr>
<td align="right"><font face="Arial">JDBC DRIVER:</font></td>
<td><font><input type="text" size="50" name="driver" value=
"sun.jdbc.odbc.JdbcOdbcDriver" /></font></td>
</tr>

<tr>
<td align="right"><font face="Arial">JDBC URL: </font> </td>
<td><font><input type="text" size="50" name="jdbcurl" value=
"jdbc:odbc:clif" /></font></td>
</tr>

<tr>
<td align="right"><font face="Arial">UserID: </font> </td>
<td><font><input type="text" size="50" name="uid" /></font></td>
</tr>

<tr>
<td align="right"><font face="Arial">Password: </font> </td>
<td><font><input type="text" size="50" name="pwd" /></font></td>
</tr>

<tr>
<td align="right"><font face="Arial">SQL Statement: </font> </td>
<td><textarea name="sql" row="10" cols="50" rows="10">Select *
From Patient</textarea></td>
</tr>
```

```
<tr>
<td align="right"><input type="Submit" /></td>
<td> </td>
</tr>

</table>

</form>

</body>

</html>
```

APPENDIX F. WEB PUBLISHING FRAMEWORK'S SOURCE CODE USING COCOON FOR GENERIC DATABASE ACCESS

A . INTRODUCTION

The Web Publishing Framework prototype site uses XML and Java technologies in the Web site is design. Section B is the ESQL Schema. Section C is GenericESQL Stylesheet.

B. ESQL SCHEMA

```
<?xml version="1.0"?>
<!-- edited with XML Spy v4.0 U (http://www.xmlspy.com) by Clifton Williams (Naval
Postgraduate School) -->
<!DOCTYPE schema SYSTEM "../../xdocs/dtd/XMLSchema.dtd">
<schema xmlns="http://www.w3.org/2000/10/XMLSchema"
xmlns:esql="http://apache.org/cocoon/SQL/v2">
    <element name="esql:ref">
        <annotation>
            <documentation>Opens a new database connection.</documentation>
        </annotation>
        <complexType>
            <sequence>
                <choice>
                    <sequence>
                        <!-- not schema strings - xsp strings -->
                        <element name="esql:driver" type="string">
                            <annotation>
                                <documentation>The name of the driver to
use</documentation>
                            </annotation>
                        </element>
                        <element name="esql:dburl" type="string">
                            <annotation>
                                <documentation>The URL of the database</documentation>
                            </annotation>
                        </element>
                        <element name="esql:username" type="string" minOccurs="0"
maxOccurs="1">
                            <annotation>
                                <documentation>The database user's name</documentation>
                            </annotation>
                        </element>
                        <element name="esql:password" type="string" minOccurs="0"
maxOccurs="1">
```

```
                     <annotation>
                        <documentation>The database user's
password</documentation>
                     </annotation>
                  </element>
               </sequence>
               <element name="esql:pool" type="string">
                  <annotation>
                     <documentation>The name of the database pool</documentation>
                  </annotation>
               </element>
            </choice>
            <sequence>
               <element name="esql:execute-query" minOccurs="0"
maxOccurs="unbounded">
                  <annotation>
                     <documentation>Executes a query on the
database</documentation>
                  </annotation>
                  <complexType>
                     <sequence>
                        <element name="esql:query">
                           <annotation>
                              <documentation>The query to execute</documentation>
                           </annotation>
                           <complexType mixed="true">
                              <sequence>
                                 <element name="esql:parameter" minOccurs="0"
maxOccurs="unbounded">
                                    <annotation>
                                       <documentation>A parameter for a prepared
statement</documentation>
                                    </annotation>
                                 </element>
                              </sequence>
                           </complexType>
                        </element>
                        <element name="esql:results" minOccurs="0" maxOccurs="1">
                           <annotation>
                              <documentation>This element's children are instantiated
in the result tree when the query returns a resultset</documentation>
                           </annotation>
                           <complexType mixed="true">
                              <sequence>
                                 <element name="esql:row-results" minOccurs="0"
maxOccurs="1">
                                    <annotation>
                                       <documentation>This element's children are
instantiated in the result tree for each row in the resultset</documentation>
                                    </annotation>
                        <complexType mixed="true">
                          <sequence>
                            <choice>
                              <any namespace="##other"
minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
                              <element name="esql:get-columns"
minOccurs="0" maxOccurs="unbounded">
                                <annotation>
                                  <documentation>Returns a
sequence of elements whose names are the current row's column names, and whose
values are the current row's column values</documentation>
                                </annotation>
                                <element name="esql:encoding"
minOccurs="0" maxOccurs="1">
                                  <annotation>
                                    <documentation>The encoding
used for this column, specified as a Java encoding
name.</documentation>
                                  </annotation>
                                </element>
                              </element>
                              <element name="esql:get-string"
minOccurs="0" maxOccurs="unbounded" type="EsqlColumnFunction">
                                <annotation>
                                  <documentation>Returns the value
of the given column in the current row as a string</documentation>
                                </annotation>
                                <element name="esql:encoding"
minOccurs="0" maxOccurs="1">
                                  <annotation>
                                    <documentation>The encoding
used for this column, specified as a Java encoding
name.</documentation>
                                  </annotation>
                                </element>
                              </element>
                              <element name="esql:get-date"
minOccurs="0" maxOccurs="unbounded" type="EsqlFormattedColumnFunction">
                                <annotation>
                                  <documentation>Returns the value
of the given column in the current row as a date</documentation>
                                </annotation>
                              </element>
                              <element name="esql:get-time"
minOccurs="0" maxOccurs="unbounded">
                                <annotation>
                                  <documentation>Returns the value
of the given column in the current row as a time</documentation>
                                </annotation>
                              </element>
```

```
                                        <element name="esql:get-timestamp"
minOccurs="0" maxOccurs="unbounded">
                                           <annotation>
                                              <documentation>Returns the value
of the given column in the current row as a timestamp</documentation>
                                           </annotation>
                                        </element>
                                        <element name="esql:get-boolean"
minOccurs="0" maxOccurs="unbounded">
                                           <annotation>
                                              <documentation>Returns the value
of the given column in the current row as a boolean</documentation>
                                           </annotation>
                                        </element>
                                        <element name="esql:get-double"
minOccurs="0" maxOccurs="unbounded">
                                           <annotation>
                                              <documentation>Returns the value
of the given column in the current row as a double</documentation>
                                           </annotation>
                                        </element>
                                        <element name="esql:get-float"
minOccurs="0" maxOccurs="unbounded">
                                           <annotation>
                                              <documentation>Returns the value
of the given column in the current row as a float</documentation>
                                           </annotation>
                                        </element>
                                        <element name="esql:get-int"
minOccurs="0" maxOccurs="unbounded">
                                           <annotation>
                                              <documentation>Returns the value
of the given column in the current row as a int</documentation>
                                           </annotation>
                                        </element>
                                        <element name="esql:get-long"
minOccurs="0" maxOccurs="unbounded">
                                           <annotation>
                                              <documentation>Returns the value
of the given column in the current row as a long</documentation>
                                           </annotation>
                                        </element>
                                        <element name="esql:get-short"
minOccurs="0" maxOccurs="unbounded">
                                           <annotation>
                                              <documentation>Returns the value
of the given column in the current row as a short</documentation>
                                           </annotation>
                                        </element>
                                        <element name="esql:get-xml"
```

```
minOccurs="0" maxOccurs="unbounded">
                                <annotation>
                                  <documentation>Parses the value
of the given column in the current row as XML and returns it</documentation>
                                </annotation>
                              </element>
                              <element name="esql:get-row-position"
minOccurs="0" maxOccurs="unbounded">
                                <annotation>
                                  <documentation>Returns the
position of the current row in the resultset</documentation>
                                </annotation>
                              </element>
                              <element name="esql:get-column-name"
minOccurs="0" maxOccurs="unbounded">
                                <annotation>
                                  <documentation>Returns the name
of the given column in the current row</documentation>
                                </annotation>
                              </element>
                              <element name="esql:get-column-label"
minOccurs="0" maxOccurs="unbounded">
                                <annotation>
                                  <documentation>Returns the label
of the given column in the current row</documentation>
                                </annotation>
                              </element>
                              <element name="esql:get-column-type-
name" minOccurs="0" maxOccurs="unbounded">
                                <annotation>
                                  <documentation>Returns the type
name of the given column in the current row</documentation>
                                </annotation>
                              </element>
                            </choice>
                          </sequence>
                        </complexType>
                      </element>
                      <element name="esql:no-results" minOccurs="0"
maxOccurs="1">
                        <annotation>
                          <documentation>This element's children are
instantiated in the result tree for each row in the resultset</documentation>
                        </annotation>
                      </element>
                    </sequence>
                  </complexType>
                </element>
                <element name="esql:update-results" minOccurs="0"
maxOccurs="1">
```

```
                                    <annotation>
                                        <documentation>This element's children are instantiated
in the result tree if the query returns a number of rows</documentation>
                                    </annotation>
                                </element>
                                <element name="esql:error-results" minOccurs="0"
maxOccurs="1">
                                    <annotation>
                                        <documentation>This element's children are instantiated
in the result tree if the query results in a database error. If this element does not exist,
the error is passed on to the xsp layer. If it does exist, the error is caught here unless
passed on explicitly.</documentation>
                                    </annotation>
                                </element>
                            </sequence>
                        </complexType>
                    </element>
                </sequence>
            </sequence>
        </complexType>
    </element>
    <complexType name="EsqlColumnFunction">
        <annotation>
            <documentation>The given column is indicated by the column attribute. If the
attribute's value is an integer, it is treated as the index of the column in the row,
otherwise it's value is treated as the name of the column.</documentation>
        </annotation>
        <attribute name="column" type="string"/>
    </complexType>
    <complexType name="EsqlFormattedColumnFunction">
        <annotation>
            <documentation>The value of the format attribute can used by some column
value functions (e.g. date, real) as a template for formatting the column value. This is
necessarily implementation dependent.</documentation>
        </annotation>
        <complexContent>
            <extension base="EsqlColumnFunction">
                <attribute name="format" type="string"/>
            </extension>
        </complexContent>
    </complexType>
</schema>
```

C. GENERICESQL STYLESHEET

```
<?xml version="1.0"?>
<!-- developed using XML Spy v4.0 U (http://www.xmlspy.com) by Clifton Williams
(Naval Postgraduate School) -->

<xsl:stylesheet
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
 version="1.0"
>

<xsl:template match="/page">
 <html>
  <head>
  <title>
      This is Clif's generic Database connection using Cocoon
</title>
</head>
<body>
<p><img border="0" src="http://www.xmlking.org/images/NPS_r01_c1.gif" alt="NPS
Header" width="969" height="90" /></p>
<palign="center"><img border="0" src="http://www.xmlking.org/images/Flag.gif"
alt="American Flag" align="left" /> <img border="0"
src="http://www.xmlking.org/images/HerrmannHallHomePage.jpg" alt="Herrmann Hall"
width="266" height="183" /></p>
   <p>
     <xsl:value-of select="header"/>
    </p>
    <table border="1" cellspacing="0" cellpadding="5" align="center">
     <tr bgcolor="#FFD700">

      <td><b>PatientID</b></td>
      <td><b>FName</b></td>
      <td><b>LName</b></td>
      <td><b>Email</b></td>
      <td><b>Gender</b></td>
      <td><b>Title</b></td>
      <td><b>Address</b></td>
      <td><b>City</b></td>
      <td><b>State</b></td>
      <td><b>Postal Code</b></td>
      <td><b>Country</b></td>
      <td><b>Home Phone</b></td>
   </tr>
```

```
    <xsl:for-each select="department">
          <tr>
            <td>
                 <xsl:value-of select="PatientID"/>
            </td>
            <td>
                 <xsl:value-of select="FName"/>
            </td>
            <td>
                 <xsl:value-of select="LName"/>
            </td>
             <td>
                <xsl:value-of select="Email"/>
            </td>
            <td>
              <xsl:value-of select="gender"/>
            </td>
            <td>
              <xsl:value-of select="Title"/>
            </td>
             <td>
                 <xsl:value-of select="Address"/>
            </td>
            <td>
                 <xsl:value-of select="City"/>
            </td>
            <td>
                 <xsl:value-of select="State"/>
            </td>
             <td>
                <xsl:value-of select="PostalCode"/>
            </td>
            <td>
              <xsl:value-of select="Country"/>
            </td>
            <td>
              <xsl:value-of select="HomePhone"/>
            </td>
          </tr>
        </xsl:for-each>
      </table>
      <p>
        <xsl:value-of select="footer"/>
      </p>
    </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

APPENDIX G. WEB PUBLISHING FRAMEWORK'S SOURCE CODE USING COCOON FOR X3D ENVIRONMENTS

A . INTRODUCTION

The Web Publishing Framework prototype site uses XML and Java technologies in the Web site is design. Section B is the X3DCocoon Servlet. Section C is the X3DVRML.XSL used by the Framework to convert Coc"XMLfile".XML to VRML. Section D is the X3D XML file HelloWorld.XML. Section E is the X3DtoHTML.XSL used by the X3dCocoon Servlet to convert .XML to HTML. Section F is the X3DtoHTML output Helloworld.HTML. Section G is the XML output CocHelloWorld.xml used by the Framework, which is converted to VRML. Section H is the XML output CocKelpForest.xml used by the Framework, which is converted to VRML. Section I is the XML output CocAmphibiousRaid.xml used by the Framework which is converted to VRML. Section J is the XHTML X3DCocoon Selection source code. Section K is only a small part of X3DtoVrml97.XSL used by the X3dCocoon Servlet to convert .XML to VRML. The Full verision of the X3DtoVrml97.XSL stylesheet can be obtained at [X3D].

B. X3D COCOON CLASS

```
package jdbc.xml;

import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import org.xml.sax.SAXException;
import org.apache.xalan.xslt.*;


/**
 * Title:         X3dCocoon CLASS
 * Description:  The Class encapsulates the functionality convert X3D Document
 *               into HTML display, VRML display, or a XML.document which is
 *               automatically loaded in the web publishing framework to be place
 *               Cocoon VRML directory for web publishing framework interface.
 *
 *
 * Company:      www.xmlking.org
 * @author       Clifton Williams
 * @version      1.6
 */

public class X3dCocoon extends HttpServlet
{
   private String qryPath;

   private String xmlFile;

   private String xslFile;

   private final int INCOMPLETE_DATA = 0;

   private final int CANNOT_ACCESS = 1;

   private final int DATA_NOT_FOUND = 2;

   private HttpServletResponse res;

   private HttpServletRequest  req;

   private PrintWriter out;

   private Writer xmlout;

   /**
```

```
    *  Process the xml document and xslt stylesheet and generates the user selected
    *  output. HTML VRML or XML Document (Cocoon Wrapper)
    */
   public void doGet(HttpServletRequest req, HttpServletResponse res)
     throws IOException, ServletException
  {

     this.res = res;
     this.req = req;
     out = res.getWriter();

     boolean badData = false;

     //get the user parameter
     xmlFile= req.getParameter("xmlInput");
     xslFile = req.getParameter("xsltInput");

     //checks to see if any entry has been made
     if(xmlFile.length() == 0 ||
         xslFile.length() == 0)
     {
        //Sets the Content Type
        res.setContentType("text/html");
        processError(INCOMPLETE_DATA );
     }//end if
     else
     {
        String xsltSystemId = new File("C:/jakarta-tomcat-3.2.3/webapps/jdbcxml/WEB-
INF/classes/jdbc/xml/"+ xslFile + "" ).toURL().toExternalForm();
        //checks path
        if(req.getPathInfo()==null)
        {
           qryPath = getServletConfig().getServletContext().getRealPath(req.getServletPath());
        }//end if
        else
        {
           qryPath = req.getPathTranslated();
        }//end else
        String output = new File("C:/jakarta-tomcat-3.2.3/webapps/jdbcxml/WEB-
INF/classes/jdbc/xml/"+ xmlFile + "").toURL().toExternalForm();
        try
        {
           //process the xml document with xsl document generates output for browser
           XSLTProcessor proc = XSLTProcessorFactory.getProcessor();
           proc.process(new XSLTInputSource((output)),
              new XSLTInputSource(xsltSystemId), new XSLTResultTarget(out));
        }//end try
        catch(SAXException se)
        {
             throw new ServletException(se);
        }//end catch
        //checks the stylesheet name and applies the correct Content type
        if (xslFile.equalsIgnoreCase("X3dVRML.xsl"))
          {   // begin anchor tag set
              res.setContentType("model/vrml");
          }   // end if
          else if(xslFile.equalsIgnoreCase("X3dCocoon.xsl"))
          {
          //This is used to generate a cocoon wrapper to run in the cocoon framework
             try{
                 //generates a Cocoon wrapper the the web publishing framework
                 FileWriter fileOut = new FileWriter("C:/jakarta-tomcat-
3.2.3/webapps/cocoon/samples/vrml/thesis/Coc"+xmlFile,false);
                 BufferedWriter bw = new BufferedWriter(fileOut);
                 bw.write("<?xml-stylesheet href=\"x3dvrml.xsl\" type=\"text/xsl\"?>");
                 bw.write("<?cocoon-process type=\"xslt\"?>");
                 bw.write("<page>\r\n");
                 bw.write("<title>\r\n "+ xmlFile +" with Cocoon </title>\r\n");
                 bw.write("<vrml>\r\n");
                 bw.close();
                 //appends the vrml output to the file
                 fileOut = new FileWriter("C:/jakarta-tomcat-
3.2.3/webapps/cocoon/samples/vrml/thesis/Coc"+xmlFile,true);
                 try
                 {
                    //process the xml document with xsl document generates output for browser
                   xslFile = “X3DToVrml97.xsl”;
                     xsltSystemId = new File("C:/jakarta-tomcat-3.2.3/webapps/jdbcxml/WEB-
INF/classes/jdbc/xml/"+ xslFile + "" ).toURL().toExternalForm();
                    XSLTProcessor proc = XSLTProcessorFactory.getProcessor();
                    proc.process(new XSLTInputSource((output)),
                      new XSLTInputSource(xsltSystemId), new XSLTResultTarget(fileOut));
                 }//end try
                 catch(SAXException se)
                 {
                  throw new ServletException(se);
                 }//end catch
                  bw.close();
                 //Stores the xml file in the cocoon web publishing work appends to file
                 fileOut = new FileWriter("C:/jakarta-tomcat-
3.2.3/webapps/cocoon/samples/vrml/thesis/Coc"+xmlFile,true);
                 bw = new BufferedWriter(fileOut);
                 bw.write("</vrml>\r\n");
                 bw.write("</page>\r\n");
                 bw.close();
              }//end try
              catch(FileNotFoundException fileEx)
              {
                  errorMsg(fileEx.getMessage());
                  processError( CANNOT_ACCESS);
              }//end catch
              catch(IOException IOex)
```

```
            {
                errorMsg(IOex.getMessage());
                processError( CANNOT_ACCESS);
            }// end catch
        }//end elseif
        else
        {
           //default setting to content type
           res.setContentType("text/html");
        }//end else
   }//end else
}//end method

/**
 *    Process the Http servlet request and response
 */
public void doPost(HttpServletRequest req, HttpServletResponse res)
   throws IOException, ServletException
{
doGet(req,res);
}//end method

/**
 * processes Errors received in verify User and generate an DHTML in response
 */
 private void processError( int errorType )
 {
    String errorMsg = "";

    switch ( errorType ) {

      case INCOMPLETE_DATA:

          errorMsg = "Please provide data to all fields";
```

```
            break;

        case CANNOT_ACCESS:

            errorMsg = "Error: Writing access file.\n" + errorMsg;
            break;

         case DATA_NOT_FOUND:

            errorMsg = "No match was found. Please try agian";
            break;
      }//end switch

      //Generate Error the title
      out.println("<html><head><title>");
      out.println("Error Has Occurred" );
      out.println("</title></head>");

      //Generates the body
      out.println("<body><h1>");
      out.println("Cannot validate you for the following reason: " );
      out.println("</h1> " );
      out.println("<h3>");
      out.println(errorMsg );
      out.println("<br></br>");

      //links to try to login again HTML
      out.println("<p><h3>");
      out.println("<form method=\" POST\" action=\"http://www.xmlking.org/X3dSelection.html\" >");
      out.println("<left><br>Try Again");
      out.println("<input type =\"submit\" value=\"Try Again\" name=\"B1\"style=\"background-
color: #006666; color: #FFFFFF; border-style: ridge; border-color: #006666\">");
      out.println("</br></left></form>");
      out.println("</p></h3>");
      out.println("</body>");
      out.close();
   }//end process error method
 /**
  * error generating message section
  */

   public void errorMsg(String error){

      out.println("<P>The following error has occured: /n");
      out.println(error);
      out.println("</p>");

      try
      {
         wait();
      }//end try
      catch(Exception e){
      }//end catch
    }//end method
}//end class
```

C. X3DVRML.XSL

```
     <?xml version="1.0"?>
     <!—written with XML Spy v4.0 U (http://www.xmlspy.com) by Clifton Williams (Naval
Postgraduate School) -->
     <?xml-stylesheet type="text/xsl" href="C:\jakarta-tomcat-
3.2.3\webapps\cocoon\samples\vrml\X3Dvrml.xsl"?>
     <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/XSL/Transform/1.0">
```

```
       <xsl:template match="page">
       <!--  <xsl:processing-instruction  name="cocoon-format">type="model/vrml"</xsl:processing-
instruction> -->

         <xsl:processing-instruction       name="cocoon-format">type="model/vrml"</xsl:processing-
instruction>
         <!-- due to a DOM limitation, you must wrap your generated VRML
              with a fake tag, here uses <vrml>, that is stripped out by the
              VRML formatter -->
         <vrml>
           <xsl:text>#VRML V2.0 utf8</xsl:text>
                 <xsl:apply-templates select="vrml"/>
             </vrml>
      </xsl:template>


      <xsl:template match="vrml">
       <xsl:value-of select="."/>
       </xsl:template>

      </xsl:stylesheet>
```

D. HELLOWORLD.XML

```
      <?xml version="1.0" encoding="utf-8"?>
      <!--  edited  with  XML  Spy  v4.0  U  (http://www.xmlspy.com)  by  Clifton  Williams  (Naval
Postgraduate School) -->

      <!DOCTYPE X3D PUBLIC "http://www.web3D.org/TaskGroups/x3d/translation/x3d-compact.dtd"
       "file:///C:/www.web3D.org/TaskGroups/x3d/translation/x3d-compact.dtd" [
       <!ENTITY % BaseProfile "INCLUDE">
```

```
  <!ENTITY % CoreProfile "IGNORE">
  <!ENTITY % DisJavaVrmlProfile "INCLUDE">
  <!ENTITY % GeoVrmlProfile "INCLUDE">
  <!ENTITY % HAnimProfile "INCLUDE">
]>
<?xml-stylesheet type="text/xsl" href="X3dHTML.xsl"?>
<?xml-stylesheet type="text/xsl" href="X3dVRML.xsl"?>
<X3D>
  <head>
    <meta content="HelloWorld.xml" name="filename"/>
    <meta content="Simple X3D example" name="description"/>
    <meta content="30 October 2000" name="revised"/>
    <meta content="Don Brutzman" name="author"/>
    <meta content="http://www.web3D.org/TaskGroups/x3d/translation/examples/HelloWorld.xml" name="url"/>
    <meta content="X3D-Edit, http://www.web3d.org/TaskGroups/x3d/translation/README.X3D-Edit.html" name="generator"/>
  </head>
  <Scene>
    <!--Example scene to illustrate X3D tags and attributes.-->
    <Group>
      <Viewpoint description="XML Presentation!" orientation="0 1 0 1.57" position="9 -1 0"/>
      <NavigationInfo type="&#34;EXAMINE&#34; &#34;ANY&#34;"/>
      <Shape>
        <Sphere DEF="S"/>
        <Appearance>
          <ImageTexture url="&#34;earth-topo.png&#34;&#xA;&#34;earth-topo-small.gif&#34;&#xA;&#34;C://www.web3D.org/TaskGroups/x3d/translation/examples/earth-topo.png&#34;&#xA;&#34;C://www.web3D.org/TaskGroups/x3d/translation/examples/earth-topo-small.gif&#34;"/>
        </Appearance>
      </Shape>
      <Transform rotation="0 1 0 1.57" translation="0 -2 1.25">
```

```
       <Shape>
         <Text string="&#34;XML Presentation!&#34;"/>
         <Appearance>
           <Material diffuseColor="0.1 0.5 1"/>
         </Appearance>
       </Shape>
     </Transform>
   </Group>
 </Scene>
</X3D>
```

E. X3DTOHTML.XSL

```
<?xml version='1.0'?>
 <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:saxon="http://icl.com/saxon" saxon:trace="yes"
version="1.0">

<!--  XSL namespaces are in transition!  Tools are slow to catch up.
    *** Edit the topmost stylesheet tag on line 2 of this file to match the xmlns namespace
URI for your XSL tool. ***
 W3C:
 Saxon:           <xsl:stylesheet xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
 IBM XSLEditor:   <xsl:stylesheet xmlns:xsl='http://www.w3.org/XSL/Transform/1.0'>
 IE 5:            <xsl:stylesheet xmlns:xsl='http://www.w3.org/TR/WD-xsl'>
 XT:              <xsl:stylesheet xmlns:xsl='http://www.w3.org/XSL/Transform'>
-->

<!--
  <head>
   <meta name="filename"    content="X3dToHtml.xsl" />
   <meta name="author"      content="Don Brutzman" />
   <meta name="revised"     content="3 July 2001" />
```

```
        <meta name="description" content="XSL stylesheet to convert X3D source into an easily
readable HTML page." />
        <meta                                                                    name="url"
content="http://www.web3D.org/TaskGroups/x3d/translation/X3dToHtml.xsl" />
       </head>

     Recommended tool:

     -   SAXON   XML   Toolkit   (and   Instant   Saxon)   from   Michael   Kay   of   ICL,
http://users.iclway.co.uk/mhkay/saxon

     - Can also be used with Apache server

     -->

     <!-- Problems and bugs:

      - autocorrection of String, array field delimiters in ProtoInstances (GeoVrmlExample1)


       -->

     <xsl:strip-space elements="*" />
     <xsl:output method="text" encoding="utf-8" media-type="text/xsl" indent="yes" cdata-section-
elements="Script"/>
     <!-- omit-xml-declaration="no" -->




     <!-- ****** root:  start of file ****** -->
     <!-- ****** root:  start of file ****** -->
     <xsl:template match="/">
```

```
      <!-- <xsl:text>&lt;!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN&quot;
&quot;http://www.w3.org/TR/REC-html40/strict.dtd&quot;&gt;&#10;&#10;</xsl:text> -->
      <!--<xsl:processing-instruction name="cocoon-format">type="text/xsl"</xsl:processing-
instruction>-->

      <xsl:text>&lt;html&gt;&#10;</xsl:text>
      <xsl:text>&lt;head&gt;&#10;</xsl:text>
      <xsl:text>&lt;title&gt;</xsl:text>
      <xsl:variable name="fileName" select="//head/meta[@name='filename']/@content" />
      <xsl:choose>
        <xsl:when test="$fileName and $fileName!='*enter filename here*' ">
          <xsl:value-of select="//head/meta[@name='filename']/@content"/>
          <xsl:text> (X3dToHtml)</xsl:text>
         </xsl:when>
         <xsl:otherwise><xsl:text> X3dToHtml </xsl:text></xsl:otherwise>
      </xsl:choose>
      <xsl:text>&lt;/title&gt;&#10;</xsl:text>
      <xsl:text>&lt;meta name=&quot;generator&quot; content=&quot;X3dToHtml.xsl,
http://www.web3D.org/TaskGroups/x3d/translation/X3dToHtml.xsl&quot;&gt;&#10;</xsl:text>
      <xsl:text>&lt;/head&gt;&#10;</xsl:text>
      <xsl:text>&lt;body&gt;&#10;</xsl:text>
      <xsl:text>&lt;pre&gt;&#10;</xsl:text>
      <xsl:text>&amp;lt;?xml version="1.0" encoding="UTF-8"?&amp;gt;&#10;</xsl:text>
      <xsl:variable name="wrapped" select="//*[local-name()='appearance' or
       local-name()='children' or local-name()='choice' or
       local-name()='color' or local-name()='coord' or local-name()='fontStyle' or
       local-name()='geometry' or local-name()='material' or local-name()='normal' or
       local-name()='source' or local-name()='level' or local-name()='texCoord' or
       local-name()='texture' or local-name()='textureTransform']" />
      <xsl:choose>
        <xsl:when test="$wrapped">
          <xsl:text>&amp;lt;!DOCTYPE X3D PUBLIC
"http://www.web3D.org/TaskGroups/x3d/translation/x3d-compromise.dtd"&#10;</xsl:text>
```

```
          <xsl:text> "file:///C:/www.web3D.org/TaskGroups/x3d/translation/x3d-
compromise.dtd"&amp;gt;&#10;</xsl:text>
        </xsl:when>
        <xsl:otherwise>
          <xsl:text>&amp;lt;!DOCTYPE X3D PUBLIC
"http://www.web3D.org/TaskGroups/x3d/translation/x3d-compact.dtd"&#10;</xsl:text>
          <xsl:text> "file:///C:/www.web3D.org/TaskGroups/x3d/translation/x3d-
compact.dtd"&amp;gt;&#10;</xsl:text>
        </xsl:otherwise>
      </xsl:choose>
      <xsl:text>&lt;/pre&gt;&#10;</xsl:text>
      <xsl:apply-templates/>
      <xsl:text>&#10;&lt;p&gt;&#10;</xsl:text>
      <xsl:text>&#10;&amp;lt;!-- Tag color codes:  </xsl:text>
      <xsl:if test="$wrapped">
        <xsl:text>&lt;font color=&quot;BLUE&quot;&gt; &amp;lt;field&amp;gt;&lt;/font&gt;
</xsl:text>
      </xsl:if>
      <xsl:text>&amp;lt;&lt;font color=&quot;NAVY&quot;&gt;NodeName&lt;/font&gt;</xsl:text>
      <xsl:text>&lt;font color=&quot;GREEN&quot;&gt; attribute&lt;/font&gt;=&apos;</xsl:text>
      <xsl:text>&lt;font color=&quot;TEAL&quot;&gt;value&lt;/font&gt;&apos;/&amp;gt;</xsl:text>
      <xsl:if test="$wrapped">
        <xsl:text>&lt;font color=&quot;BLUE&quot;&gt;
&amp;lt;/field&amp;gt;&lt;/font&gt;</xsl:text>
      </xsl:if>
      <xsl:text> --&amp;gt;&#10;</xsl:text>

      <xsl:text>&#10;&lt;/p&gt;&#10;</xsl:text>
      <xsl:text>&lt;/body&gt;&#10;</xsl:text>
      <xsl:text>&lt;/html&gt;&#10;</xsl:text>
      </xsl:template>

      <!-- ****** recurse through each of the tree node elements ****** -->
      <xsl:template match="*">
      <xsl:if test="local-name(..)='Script' and local-name()='field'"><xsl:text>&#10;</xsl:text></xsl:if>
      <!-- first tag name -->
      <xsl:text>&amp;lt;&lt;font color=&quot;NAVY&quot;&gt;</xsl:text>
      <xsl:value-of select="local-name()"/>
      <xsl:text>&lt;/font&gt;</xsl:text>
      <!-- next attributes, if any.  output in order if appropriate. -->
      <xsl:if test="@*">
        <xsl:choose>
          <xsl:when test="local-name()='component'">
            <xsl:apply-templates select="@name"/>
            <xsl:apply-templates select="@*[local-name()!='name']"/>
          </xsl:when>
          <xsl:when test="local-name()='meta'">
            <xsl:apply-templates select="@name"/>
            <xsl:apply-templates select="@content"/>
            <xsl:apply-templates select="@*[local-name()!='name' and local-name()!='content']"/>
          </xsl:when>
          <xsl:when test="local-name()='ROUTE'">
            <xsl:apply-templates select="@fromNode"/>
            <xsl:apply-templates select="@fromField"/>
            <xsl:apply-templates select="@toNode"/>
            <xsl:apply-templates select="@toField"/>
          </xsl:when>
          <xsl:when test="local-name()='ElevationGrid' or local-name()='GeoElevationGrid'">
            <xsl:apply-templates select="@*[local-name()!='height' and local-name()!='colorIndex']"/>
            <xsl:apply-templates select="@height"/>
            <xsl:apply-templates select="@colorIndex"/>
          </xsl:when>
          <xsl:when test="local-name()='IndexedFaceSet'">
            <xsl:apply-templates select="@*[not(contains(local-name(), 'Index'))]"/>
            <xsl:apply-templates select="@*[contains(local-name(), 'Index')]"/>
          </xsl:when>
          <xsl:when test="local-name()='field' or local-name()='fieldValue'">
            <xsl:apply-templates select="@name"/>
            <xsl:apply-templates select="@type"/>
            <xsl:apply-templates select="@value"/>
            <xsl:apply-templates select="@*[(local-name()!='name') and (local-name()!='type') and
(local-name()!='value')]"/>
          </xsl:when>
          <xsl:otherwise>
            <xsl:apply-templates select="@DEF | @USE | @*[not(contains(local-name(), 'url') or
contains(local-name(), 'Url'))]"/>
            <xsl:apply-templates select="@*[contains(local-name(), 'url') or contains(local-
name(), 'Url')]"/>
          </xsl:otherwise>
        </xsl:choose>
      </xsl:if>
      <!-- appropriate angle-bracket close of first tag, depending if children node/comment is
present -->
      <xsl:choose>
        <xsl:when test="node() |
comment()"><xsl:text>&amp;gt;&#10;&lt;blockquote&gt;</xsl:text></xsl:when>
        <xsl:when test="local-name(..)='Script' and local-
name()='field'"><xsl:text>/&amp;gt;</xsl:text></xsl:when>
        <xsl:otherwise><xsl:text>/&amp;gt;&#10;</xsl:text></xsl:otherwise>
      </xsl:choose>
      <!-- recurse on contained tag/comment -->
      <xsl:apply-templates select="node() | comment()"/>
      <!-- add a closing tag if children node/comment is present -->
      <xsl:if test="* or comment()">
        <xsl:text>&lt;/blockquote&gt;&#10;&amp;lt;/&lt;font color=&quot;NAVY&quot;&gt;</xsl:text>
        <xsl:value-of select="local-name()"/>
        <xsl:text>&lt;/font&gt;&amp;gt;&#10;</xsl:text>
      </xsl:if>
      <!-- *** need test for last -->
      <xsl:text>&lt;br&gt;&#10;</xsl:text>
```

```
    </xsl:template>

    <!-- ****** recurse through each of the attributes ****** -->
    <xsl:template match="@*">
      <!-- eliminate default attribute values, otherwise they will all appear in output  -->
      <!-- this block is used identically in X3dToVrml97.xsl X3dToHtml.xsl X3dUnwrap.xsl and
X3dWrap.xsl -->
      <!-- check values with/without .0 suffix since these are string checks and
autogenerated/DOM output might have either -->
      <!-- do not check ProtoInstances or natively defined nodes, since they might have
different user-defined defaults -->
      <!-- tool-bug workaround:  split big boolean queries into pieces to avoid overloading the
Xalan/lotusxml query buffer -->
      <xsl:variable name="notImplicitEvent"
      select="not(local-name(..)='AudioClip' and (local-name()='duration' or local-
name()='isActive')) and
        not(contains(local-name(..),'Interpolator') and (local-name()='fraction' or local-
name()='value')) and
        not(local-name(..)='Background' and (local-name()='bind' or local-name()='bindTime' or
local-name()='isBound')) and
        not(local-name(..)='CylinderSensor' and (local-name()='isActive' or local-
name()='rotation' or local-name()='trackPoint')) and
        not(local-name(..)='Fog' and (local-name()='bind' or local-name()='bindTime' or local-
name()='isBound')) and
        not(local-name(..)='IndexedLineSet' and local-name()='lineWidth') and
        not(local-name(..)='NavigationInfo' and (local-name()='bind' or local-name()='bindTime'
or local-name()='isBound')) and
        not(local-name(..)='PointSet' and local-name()='pointSize') and
        not(local-name(..)='PlaneSensor' and(local-name()='isActive' or local-
name()='translation' or local-name()='trackPoint')) and
        not(local-name(..)='ProximitySensor' and (local-name()='isActive' or local-
name()='position' or local-name()='orientation' or
                     local-name()='enterTime' or local-name()='exitTime')) and
```

```
         not(local-name(..)='SphereSensor' and (local-name()='isActive' or local-
name()='rotation' or local-name()='trackPoint')) and
         not(local-name(..)='TimeSensor' and (local-name()='isActive' or local-name()='cycleTime'
or local-name()='fraction' or
                     local-name()='time')) and
         not(local-name(..)='TouchSensor' and (local-name()='isActive' or local-name()='isOver' or
local-name()='hitNormal' or
                     local-name()='touchTime' or local-name()='hitPoint' or local-
name()='hitTexCoord')) and
         not(local-name(..)='Viewpoint' and (local-name()='bind' or local-name()='bindTime' or
local-name()='isBound' or local-name()='examine'))" />
       <xsl:variable name="notDefaultFieldValue1"
       select="not( local-name()='bboxCenter' and (.='0 0 0' or .='0.0 0.0 0.0')) and
         not( local-name()='bboxSize' and (.='-1 -1 -1' or .='-1.0 -1.0 -1.0')) and
         not( local-name(..)='AudioClip' and
                  ((local-name()='loop' and .='false') or
                   (local-name()='pitch' and (.='1' or .='1.0')) or
                   (local-name()='startTime' and (.='0' or .='0.0')) or
                   (local-name()='stopTime' and (.='0 0 0' or .='0.0 0.0 0.0')))) and
         not( local-name(..)='Background' and local-name()='skyColor' and (.='0 0 0' or .='0.0 0.0
0.0')) and
         not( local-name(..)='Billboard' and local-name()='axisOfRotation' and (.='0 1 0' or
.='0.0 1.0 0.0')) and
         not( local-name(..)='Box' and local-name()='size' and (.='2 2 2' or .='2.0 2.0 2.0'))
and
         not( local-name(..)='Collision' and local-name()='collide' and .='true') and
         not( local-name(..)='Cone' and ((local-name()='bottomRadius' and (.='1' or .='1.0')) or
                   (local-name()='height' and (.='2' or .='2.0')) or
                   (local-name()='side' and .='true') or
                   (local-name()='bottom' and .='true'))) and
         not( local-name(..)='Cylinder' and
                  ((local-name()='height' and (.='2' or .='2.0')) or
                   (local-name()='radius' and (.='1' or .='1.0')) or
                   (local-name()='bottom' and .='true') or
```

```
             (local-name()='side' and .='true') or
             (local-name()='top' and .='true'))) and
  not( local-name(..)='CylinderSensor' and
            ((local-name()='autoOffset' and .='true') or
             (local-name()='enabled' and .='true') or
             (local-name()='diskAngle' and .='0.262') or
             (local-name()='offset' and (.='0' or .='0.0')) or
             (local-name()='maxAngle' and (.='-1' or .='-1.0')) or
             (local-name()='minAngle' and (.='0' or .='0.0'))))" />
 <xsl:variable name="notDefaultFieldValue2"
select="not( local-name(..)='DirectionalLight' and
            ((local-name()='ambientIntensity' and (.='0' or .='0.0')) or
             (local-name()='color' and (.='1 1 1' or .='1.0 1.0 1.0')) or
             (local-name()='direction' and (.='0 0 -1' or .='0.0 0.0 -1.0')) or
             (local-name()='intensity' and (.='1' or .='1.0')) or
             (local-name()='on' and .='true'))) and
  not( local-name(..)='ElevationGrid' and
            ((local-name()='ccw' and .='true') or
             (local-name()='colorPerVertex' and .='true') or
             (local-name()='normalPerVertex' and .='true') or
             (local-name()='solid' and .='true') or
             (local-name()='xDimension' and (.='0' or .='0.0')) or
             (local-name()='xSpacing' and (.='1' or .='1.0')) or
             (local-name()='zDimension' and (.='0' or .='0.0')) or
             (local-name()='zSpacing' and (.='1' or .='1.0')) or
             (local-name()='creaseAngle' and (.='0' or .='0.0')))) and
  not( local-name(..)='Extrusion' and
            ((local-name()='beginCap' and .='true') or
             (local-name()='ccw' and .='true') or
             (local-name()='convex' and .='true') or
             (local-name()='endCap' and .='true') or
             (local-name()='solid' and .='true') or
             (local-name()='creaseAngle' and (.='0' or .='0.0')) or
             (local-name()='orientation' and (.='0 0 1 0' or .='0.0 0.0 1.0 0.0')) or
```

```
                   (local-name()='scale' and (.='1 1' or .='1.0 1.0')) or
                   (local-name()='crossSection' and .='1 1, 1 -1, -1 -1, -1 1, 1  1') or
                   (local-name()='spine' and .='0 0 0, 0 1 0')))" />
       <xsl:variable name="notDefaultFieldValue3"
       select="not( local-name(..)='Fog' and  ((local-name()='color' and (.='1 1 1' or .='1.0 1.0
1.0')) or
                   (local-name()='visibilityRange' and (.='0' or .='0.0')) or
                   (local-name()='fogType' and .='&quot;LINEAR&quot;'))) and
         not( local-name(..)='FontStyle' and
                  ((local-name()='horizontal' and .='true') or
                   (local-name()='leftToRight' and .='true') or
                   (local-name()='topToBottom' and .='true') or
                   (local-name()='size' and (.='1' or .='1.0')) or
                   (local-name()='spacing' and (.='1' or .='1.0')) or
                   (local-name()='family' and .='&quot;SERIF&quot;') or
                   (local-name()='justify' and .='&quot;BEGIN&quot;') or
                   (local-name()='style' and .='PLAIN'))) and
         not( local-name(..)='ImageTexture' and
                  ((local-name()='repeatS' and .='true') or
                   (local-name()='repeatT' and .='true'))) and
         not( local-name(..)='IndexedFaceSet' and
                  ((local-name()='ccw' and .='true') or
                   (local-name()='colorPerVertex' and .='true') or
                   (local-name()='convex' and .='true') or
                   (local-name()='normalPerVertex' and .='true') or
                   (local-name()='solid' and .='true') or
                   (local-name()='creaseAngle' and (.='0' or .='0.0')))) and
         not( local-name(..)='IndexedLineSet' and local-name()='colorPerVertex' and .='true') and
         not( local-name(..)='LOD ' and  local-name()='center'  and  (.='0  0  0'  or  .='0.0  0.0
0.0')) and
         not( local-name(..)='Material'  and
                  ((local-name()='ambientIntensity' and .='0.2') or
                   (local-name()='diffuseColor' and .='0.8 0.8 0.8') or
                   (local-name()='emissiveColor' and (.='0 0 0' or .='0.0 0.0 0.0')) or
```

```
             (local-name()='shininess' and .='0.2') or
             (local-name()='specularColor' and (.='0 0 0' or .='0.0 0.0 0.0')) or
             (local-name()='transparency' and (.='0' or .='0.0'))))" />
 <xsl:variable name="notDefaultFieldValue4"
select="not( local-name(..)='MovieTexture' and
            ((local-name()='loop' and .='false') or
             (local-name()='speed' and (.='1' or .='1.0')) or
             (local-name()='startTime' and (.='0' or .='0.0')) or
             (local-name()='stopTime' and (.='0' or .='0.0')) or
             (local-name()='repeatS' and .='true') or
             (local-name()='repeatT' and .='true'))) and
  not( local-name(..)='NavigationInfo' and
            ((local-name()='avatarSize' and .='0.25 1.6 0.75') or
             (local-name()='headlight' and .='true') or
             (local-name()='speed' and (.='1' or .='1.0')) or
             (local-name()='visibilityLimit' and (.='0' or .='0.0')))) and
  not( local-name(..)='PixelTexture' and
            ((local-name()='repeatS' and .='true') or
             (local-name()='repeatT' and .='true') or
             (local-name()='image' and (.='0 0 0' or .='0.0 0.0 0.0')))) and
  not( local-name(..)='PlaneSensor' and
            ((local-name()='autoOffset' and .='true') or
             (local-name()='enabled' and .='true') or
             (local-name()='maxPosition' and (.='-1 -1' or .='-1.0 -1.0')) or
             (local-name()='minPosition' and (.='0 0' or .='0.0 0.0')) or
             (local-name()='offset' and (.='0 0 0' or .='0.0 0.0 0.0')))) and
  not( local-name(..)='PointLight' and
            ((local-name()='ambientIntensity' and (.='0' or .='0.0'))or
             (local-name()='attenuation' and (.='1 0 0' or .='1.0 0.0 0.0')) or
             (local-name()='color' and (.='1 1 1' or .='1.0 1.0 1.0')) or
             (local-name()='intensity' and (.='1' or .='1.0')) or
             (local-name()='location' and (.='0 0 0' or .='0.0 0.0 0.0')) or
             (local-name()='on' and .='true') or
             (local-name()='radius' and (.='100' or .='100.0'))))" />
```

```
       <xsl:variable name="notDefaultFieldValue5"
       select="not( local-name(..)='ProximitySensor' and
                  ((local-name()='center' and (.='0 0 0' or .='0.0 0.0 0.0')) or
                   (local-name()='size' and (.='0 0 0' or .='0.0 0.0 0.0')) or
                   (local-name()='enabled' and .='true'))) and
         not( local-name(..)='Script' and ((local-name()='directOutput' and .='false') or
                   (local-name()='mustEvaluate' and .='false'))) and
         not( local-name(..)='Sound' and ((local-name()='direction' and (.='0 0 1' or .='0.0 0.0
1.0')) or
                   (local-name()='intensity' and (.='1' or .='1.0')) or
                   (local-name()='location' and (.='0 0 0' or .='0.0 0.0 0.0')) or
                   (local-name()='priority' and (.='0' or .='0.0')) or
                   (local-name()='maxBack' and (.='10' or .='10.0')) or
                   (local-name()='maxFront' and (.='10' or .='10.0')) or
                   (local-name()='minBack' and (.='1' or .='1.0'))  or
                   (local-name()='minFront' and (.='1' or .='1.0')) or
                   (local-name()='spatialize' and .='true'))) and
         not( local-name(..)='Sphere' and  local-name()='radius' and (.='1' or .='1.0')) and
         not( local-name(..)='SphereSensor' and
                  ((local-name()='autoOffset' and .='true') or
                   (local-name()='enabled' and .='true') or
                   (local-name()='offset' and (.='0 1 0 0' or .='0.0 1.0 0.0 0.0'))))" />
       <xsl:variable name="notDefaultFieldValue6"
       select="not( parent::SpotLight  and
                  ((local-name()='ambientIntensity' and (.='0' or .='0.0')) or
                   (local-name()='attenuation' and (.='1 0 0' or .='1.0 0.0 0.0')) or
                   (local-name()='beamwidth' and .='1.570796') or
                   (local-name()='color' and (.='1 1 1' or .='1.0 1.0 1.0')) or
                   (local-name()='cutOffAngle' and .='0.785398') or
                   (local-name()='direction' and (.='0 0 -1' or .='0.0 0.0 -1.0')) or
                   (local-name()='intensity' and (.='1' or .='1.0')) or
                   (local-name()='location' and (.='0 0 0' or .='0.0 0.0 0.0')) or
                   (local-name()='on' and .='true') or
                   (local-name()='radius' and (.='100' or .='100.0')))) and
         not( local-name(..)='Switch' and  local-name()='whichChoice' and (.='-1' or .='-1.0'))
and
         not( local-name(..)='Text'  and  local-name()='maxExtent' and (.='0' or .='0.0')) and
         not( local-name(..)='TextureTransform' and
                  ((local-name()='center' and (.='0 0' or .='0.0 0.0')) or
                   (local-name()='rotation' and (.='0' or .='0.0')) or
                   (local-name()='scale ' and (.='1 1' or .='1.0 1.0')) or
                   (local-name()='translation' and (.='0 0' or .='0.0 0.0')))) and
         not( local-name(..)='Switch' and  local-name()='whichChoice' and (.='-1' or .='-1.0'))"
/>
       <xsl:variable name="notDefaultFieldValue7"
      select="not( local-name(..)='TimeSensor' and
                  ((local-name()='cycleInterval' and (.='1' or .='1.0')) or
                   (local-name()='enabled' and .='true') or
                   (local-name()='startTime' and (.='0' or .='0.0')) or
                   (local-name()='stopTime' and (.='0' or .='0.0')) or
                   (local-name()='loop' and .='false'))) and
         not( local-name(..)='TouchSensor' and
                   local-name()='enabled' and .='true') and
         not( local-name(..)='Transform' and
                  ((local-name()='center' and (.='0 0 0' or .='0.0 0.0 0.0')) or
                   (local-name()='rotation' and (.='0 0 1 0' or .='0.0 0.0 1.0 0.0')) or
                   (local-name()='scale' and (.='1 1 1' or .='1.0 1.0 1.0')) or
                   (local-name()='scaleOrientation' and (.='0 0 1 0' or .='0.0 0.0 1.0 0.0')) or
                   (local-name()='translation' and (.='0 0 0' or .='0.0 0.0 0.0')))) and
         not( local-name(..)='Viewpoint' and
                  ((local-name()='fieldOfView' and .='0.785398') or
                   (local-name()='jump' and .='true') or
                   (local-name()='orientation' and (.='0 0 1 0' or .='0.0 0.0 1.0 0.0')) or
                   (local-name()='position' and (.='0 0 10' or .='0.0 0.0 10.0')))) and
         not( local-name(..)='VisibilitySensor' and
                  ((local-name()='center' and (.='0 0 0' or .='0.0 0.0 0.0')) or
                   (local-name()='enabled' and .='true') or
                   (local-name()='size' and (.='0 0 0' or .='0.0 0.0 0.0'))))" />
       <xsl:variable name="notDefaultHAnim"
       select="not( local-name(..)='Joint' and
                  ((local-name()='center' and (.='0 0 0' or .='0.0 0.0 0.0')) or
                   (local-name()='limitOrientation' and (.='0 0 1 0' or .='0.0 0.0 1.0 0.0')) or
                   (local-name()='rotation' and (.='0 0 1 0' or .='0.0 0.0 1.0 0.0')) or
                   (local-name()='scale' and (.='1 1 1' or .='1.0 1.0 1.0')) or
                   (local-name()='scaleOrientation' and (.='0 0 1 0' or .='0.0 0.0 1.0 0.0')) or
                   (local-name()='stiffness' and (.='0 0 0' or .='0.0 0.0 0.0')) or
                   (local-name()='translation' and (.='0 0 0' or .='0.0 0.0 0.0')))) and
         not( local-name(..)='Segment' and
                  ((local-name()='bboxCenter' and (.='0 0 0' or .='0.0 0.0 0.0')) or
                   (local-name()='bboxSize' and (.='-1 -1 -1' or .='-1.0 -1.0 -1.0')) or
                   (local-name()='centerOfMass' and (.='0 0 0' or .='0.0 0.0 0.0')) or
                   (local-name()='mass' and (.='0' or .='0.0')) or
                   (local-name()='momentsOfInertia' and
                    (.='0 0 0 0 0 0 0 0 0' or .='0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0')))) and
         not( local-name(..)='Site' and
                  ((local-name()='center' and (.='0 0 0' or .='0.0 0.0 0.0')) or
                   (local-name()='rotation' and (.='0 0 1 0' or .='0.0 0.0 1.0 0.0')) or
                   (local-name()='scale' and (.='1 1 1' or .='1.0 1.0 1.0')) or
                   (local-name()='scaleOrientation' and (.='0 0 1 0' or .='0.0 0.0 1.0 0.0')) or
                   (local-name()='translation' and (.='0 0 0' or .='0.0 0.0 0.0')))) and
         not( local-name(..)='Humanoid' and
                  ((local-name()='bboxCenter' and (.='0 0 0' or .='0.0 0.0 0.0')) or
                   (local-name()='bboxSize' and (.='-1 -1 -1' or .='-1.0 -1.0 -1.0')) or
                   (local-name()='center' and (.='0 0 0' or .='0.0 0.0 0.0')) or
                   (local-name()='rotation' and (.='0 0 1 0' or .='0.0 0.0 1.0 0.0')) or
                   (local-name()='scale' and (.='1 1 1' or .='1.0 1.0 1.0')) or
                   (local-name()='scaleOrientation' and (.='0 0 1 0' or .='0.0 0.0 1.0 0.0')) or
                   (local-name()='translation' and (.='0 0 0' or .='0.0 0.0 0.0'))))" />
       <xsl:variable name="notFieldSpace"
       select="not( local-name(..)='field' and   (local-name()='space'          or          local-
name()='xml:space'))" />
     <xsl:if test=" $notImplicitEvent and
```

```
         $notDefaultFieldValue1 and
         $notDefaultFieldValue2 and
         $notDefaultFieldValue3 and
         $notDefaultFieldValue4 and
         $notDefaultFieldValue5 and
         $notDefaultFieldValue6 and
         $notDefaultFieldValue7 and
                     $notDefaultHAnim and
                     $notFieldSpace" >
             <!--and not((local-name(../..)='Script' and local-name(..)='field') and (local-
name()='xml:space' or local-name()='space')) -->
       <!-- valid field found by the preceding checks, now output accordingly  -->
       <!-- single attributes can stay on same line, skip line otherwise -->
       <!-- problem:  appears to be counting default attributes in addition to user-defined
attributes... -->
       <xsl:if test="(count (../@*) > 2) or ((count (../@*) = 2) and (string-length(../@*[1]) +
string-length(../@*[2]) &gt; 72))">
         <xsl:text>&#10;  </xsl:text>
       </xsl:if>
       <xsl:text> &lt;font color=&quot;GREEN&quot;&gt;</xsl:text>
       <!-- output actual attribute value.  try to break MFStrings into multiple lines. -->
       <xsl:value-of select="local-name()"/>
       <xsl:text>&lt;/font&gt;&lt;B&gt;=&apos;&lt;/B&gt;</xsl:text>
       <xsl:choose>
         <!-- print out urls containing javascript source without further ado -->
         <xsl:when test="(local-name()='url' and starts-with(normalize-space(.),'javascript:'))">
           <xsl:text>&lt;PRE&gt;</xsl:text>
           <xsl:value-of select="." disable-output-escaping="yes"/>
           <xsl:text>&lt;/PRE&gt;</xsl:text>
         </xsl:when>
         <!-- make single url reference into actual A HREF= link -->
         <xsl:when test="(local-name()='url' or contains(local-name(), 'Url') or (local-
name(..)='meta' and (../@name='url' or ../@name='filename' or ../@name='reference' or
../@name='copyright' or ../@name='drawing' or ../@name='image' or ../@name='movie' or
../@name='photo' or ../@name='photograph' or contains(../@name,'exture')) and local-
name()='content')) and not(starts-with(normalize-space(.),'&quot;')) and (contains(.,'.') or
starts-with(normalize-space(.),'Makefile')) and not(substring(normalize-space(.),string-
length(normalize-space(.))) = '.')">
            <xsl:text>&lt;a href=&quot;</xsl:text><xsl:value-of select="normalize-space(.)"
disable-output-escaping="yes"/><xsl:text>&quot;&gt;</xsl:text><xsl:value-of select="normalize-
space(.)" disable-output-escaping="yes"/><xsl:text>&lt;/a&gt;</xsl:text></xsl:when>
          <xsl:when test="(local-name()='url' or contains(local-name(), 'Url') or (local-
name(..)='meta' and (../@name='url' or ../@name='filename' or ../@name='reference' or
../@name='drawing' or ../@name='image' or ../@name='movie' or ../@name='photo' or
../@name='photograph') and local-name()='content')) and starts-with(normalize-space(.),'&quot;')
and not(contains(normalize-space(.),'&quot; &quot;')) and (contains(.,'.') or starts-
with(normalize-space(.),'Makefile')) and not(substring(normalize-space(.),string-length(normalize-
space(.))) = '.')">
            <xsl:text>&lt;a href=</xsl:text><xsl:value-of select="normalize-space(.)" disable-
output-escaping="yes"/><xsl:text>&gt;</xsl:text><xsl:value-of select="substring-before(substring-
after(normalize-space(.),'&quot;'),'&quot;')" disable-output-
escaping="yes"/><xsl:text>&lt;/a&gt;</xsl:text></xsl:when>
          <!-- break and make multiple url references into actual A HREF= links -->
          <xsl:when test="(local-name()='url' or contains(local-name(), 'Url') or (local-
name(..)='meta' and (../@name='url' or ../@name='filename' or ../@name='reference' or
../@name='drawing' or ../@name='image' or ../@name='movie' or ../@name='photo' or
../@name='photograph') and local-name()='content')) and starts-with(normalize-space(.),'&quot;')
and contains(normalize-space(.),'&quot; &quot;')">
            <xsl:call-template name="URL-ize-MFString-elements">
              <xsl:with-param name="list" select="normalize-space(.)"/>
            </xsl:call-template>
          </xsl:when>
          <xsl:when test="starts-with(normalize-space(.),'http://')">
            <xsl:text>&lt;a href="</xsl:text><xsl:value-of select="normalize-space(.)" disable-
output-escaping="yes"/><xsl:text>"&gt;</xsl:text><xsl:value-of select="normalize-space(.)"
disable-output-escaping="yes"/><xsl:text>&lt;/a&gt;</xsl:text>
          </xsl:when>
          <xsl:when test="(local-name(..)='meta' and local-name()='content') and
(../@name='drawing' or ../@name='image' or ../@name='movie' or ../@name='photo' or
../@name='photograph')">
            <xsl:text>&lt;a href="</xsl:text><xsl:value-of select="normalize-space(.)" disable-
output-escaping="yes"/><xsl:text>"&gt;</xsl:text><xsl:value-of select="normalize-space(.)"
disable-output-escaping="yes"/><xsl:text>&lt;/a&gt;</xsl:text>
          </xsl:when>
          <xsl:when test="starts-with(.,'X3D-Edit, ')">
            <xsl:variable name="containedUrl" select="substring-after(.,'X3D-Edit, ')"/>
            <xsl:text>X3D-Edit, </xsl:text>
            <xsl:text>&lt;a href="</xsl:text><xsl:value-of select="$containedUrl" disable-output-
escaping="yes"/><xsl:text>" target="_blank"&gt;</xsl:text><xsl:value-of select="$containedUrl"
disable-output-escaping="yes"/><xsl:text>&lt;/a&gt;</xsl:text>
          </xsl:when>
          <xsl:when test="(local-name(..)='meta' and local-name()='content' and
../@name='generator') and (contains(.,'.') and (not(contains(.,' '))))">
            <xsl:text>&lt;a href="</xsl:text><xsl:value-of select="normalize-space(.)" disable-
output-escaping="yes"/><xsl:text>"&gt;</xsl:text><xsl:value-of select="normalize-space(.)"
disable-output-escaping="yes"/><xsl:text>&lt;/a&gt;</xsl:text>
          </xsl:when>
          <xsl:when test="local-name()='url'">
            <xsl:text>&lt;font color=&quot;BLACK&quot;&gt;</xsl:text><xsl:value-of
select="normalize-space(.)" disable-output-escaping="yes"/>&lt;/font&gt;</xsl:when>
          <!-- meta warning -->
          <xsl:when test="local-name(..)='meta' and local-name()='name' and .='warning'">
            <!-- remove extraneous space for better formatting, even though this might obscure
some content problems/preferences. -->
            <xsl:text>&lt;B&gt;&lt;U&gt;&lt;font
color=&quot;ORANGE&quot;&gt;</xsl:text><xsl:value-of select="normalize-
space(.)"/>&lt;/font&gt;&lt;/U&gt;&lt;/B&gt;</xsl:when>
          <xsl:otherwise>
            <!-- remove extraneous space for better formatting, even though this might obscure
some content problems/preferences. -->
```

```
            <xsl:text>&lt;font color=&quot;TEAL&quot;&gt;</xsl:text><xsl:value-of
select="normalize-space(.)"/>&lt;/font&gt;</xsl:otherwise>
        </xsl:choose>
        <xsl:text>&lt;B&gt;&apos;&lt;/B&gt;</xsl:text>
        <xsl:if test="contains(local-name(),
'Url')"><xsl:text>&lt;br&gt;&#10;</xsl:text></xsl:if><!--  -->
      </xsl:if>
      </xsl:template>

      <!-- ****** URL-ize-MFString-elements:  callable template (recursive function) ****** -->
      <!-- follows examples in Michael Kay's _XSLT_, pp. 551-554 -->
      <!-- this will need modification if SFURL/MFURL types are created -->
      <xsl:template name="URL-ize-MFString-elements">
        <xsl:param name="list"/>
        <xsl:variable name="wlist" select="concat(normalize-space($list),' ')"/>
        <!--   <xsl:text>&#10;$wlist=[</xsl:text><xsl:value-of select="$wlist" disable-output-
escaping="yes"/><xsl:text>]&#10;</xsl:text> -->
        <xsl:choose>
          <xsl:when test="$wlist!=' '">
            <xsl:variable name="nextURL" select="translate(substring-before($wlist,'
'),'&quot;','')"/>
            <xsl:variable name="restURLs" select="substring-after($wlist,' ')"/>
            <!--  <xsl:text>&#10;$restURLs=[</xsl:text><xsl:value-of select="$restURLs" disable-
output-escaping="yes"/><xsl:text>]&#10;</xsl:text> -->
            <!-- output URL-ized nextURL -->
            <xsl:text>&quot;&lt;A href=&quot;</xsl:text><xsl:value-of select="normalize-
space($nextURL)"/><xsl:text>&quot;&gt;</xsl:text><xsl:value-of select="normalize-
space($nextURL)"/><xsl:text>&lt;/A&gt;&quot;</xsl:text>
            <!-- recurse on remainder of list of URLs -->
            <xsl:if test="$restURLs!=''"><xsl:text>&#10;</xsl:text></xsl:if>
            <xsl:call-template name="URL-ize-MFString-elements">
              <xsl:with-param name="list" select="$restURLs"/>
            </xsl:call-template>
          </xsl:when>
        </xsl:choose>
      </xsl:template>

      <!-- ****** children:  flag wrapper tags ****** -->
      <xsl:template match="appearance[parent::Shape] |
         children[parent::Anchor or parent::Billboard or parent::Collision or parent::Group   or
parent::Transform] |
         choice[parent::Switch] |
         color[parent::ElevationGrid    or  parent::IndexedFaceSet  or  parent::IndexedLineSet  or
parent::PointSet] |
         coord[parent::IndexedFaceSet or parent::IndexedLineSet or parent::PointSet] |
         fontStyle[parent::Text] |
         geometry[parent::Shape] |
         level[parent::LOD] |
         material[parent::Appearance] |
         normal[parent::ElevationGrid or parent::IndexedFaceSet] |
         source[parent::Sound] |
         texCoord[parent::ElevationGrid or parent::IndexedFaceSet] |
         texture[parent::Appearance] |
         textureTransform[parent::Appearance]">
      <!-- first, output tag name -->
      <xsl:text>&lt;font  color=&quot;BLUE&quot;&gt;&amp;lt;</xsl:text><xsl:value-of  select="local-
name()"/>
      <!-- appropriate angle-bracket close of first tag -->
      <xsl:choose>
        <xsl:when test="not(*)"><xsl:text>/&amp;gt;</xsl:text></xsl:when>
        <xsl:otherwise><xsl:text>&amp;gt;</xsl:text></xsl:otherwise>
      </xsl:choose>
      <!-- insert warning comment after wrapper tag -->
      <!-- <xsl:comment> wrapper tag </xsl:comment> -->
      <!-- prepare format for next tag, insert blockquote if children present -->
      <xsl:choose>
        <xsl:when test="not(*)"><xsl:text>&#10;&lt;/font&gt;</xsl:text></xsl:when>
        <xsl:otherwise><xsl:text>&lt;/font&gt;&lt;blockquote&gt;</xsl:text></xsl:otherwise>
      </xsl:choose>
      <!-- recurse on next tag -->
      <xsl:apply-templates />
      <!-- add a closing tag if children were present -->
      <xsl:if test="*">
        <xsl:text>&lt;/blockquote&gt;&#10;&lt;font color=&quot;BLUE&quot;&gt;&amp;lt;/</xsl:text>
        <xsl:value-of select="local-name()"/>
        <xsl:text>&amp;gt;&lt;/font&gt;&#10;</xsl:text>
      </xsl:if>
      </xsl:template>


      <!-- ****** XML comments ****** -->
      <xsl:template match="comment()[ancestor::X3D]">
      <!--  <xsl:text>&lt;PRE&gt;</xsl:text> -->
        <xsl:text>&amp;lt;!--  </xsl:text><xsl:value-of  select="normalize-space(.)"/><xsl:text>  --
&amp;gt;</xsl:text>
      <!--   <xsl:text>&lt;/PRE&gt;</xsl:text> -->
        <xsl:text>&#10;</xsl:text>
      <!-- *** need test for last -->
      <xsl:text>&lt;br&gt;&#10;</xsl:text>
      </xsl:template>


      <!-- ****** XML processing-instruction ****** -->
      <xsl:template match="processing-instruction()">
        <xsl:text>&lt;-- </xsl:text><xsl:value-of select="."/><xsl:text> --&gt;&#10;</xsl:text>
      </xsl:template>


      <!-- ****** Script node:  preserve CDATA delimiters ****** -->
        <!-- first tag name -->
        <!-- next attributes, if any -->
      <xsl:template match="Script[text()]">
```

```
        <xsl:text>&amp;lt;&lt;font color=&quot;NAVY&quot;&gt;</xsl:text><xsl:value-of
select="local-name()"/><xsl:text>&lt;/font&gt;</xsl:text>
        <xsl:apply-templates select="@*"/>
        <xsl:text>&amp;gt;&#10;&lt;blockquote&gt;</xsl:text>
        <xsl:apply-templates select="field | comment()"/>
        <xsl:text>&lt;PRE&gt;</xsl:text>
        <xsl:text>&amp;lt;![CDATA[&#10;</xsl:text>
        <xsl:value-of select="." />
        <xsl:text>&#10;</xsl:text>
        <xsl:text>]]&amp;gt;</xsl:text>
        <xsl:text>&lt;/PRE&gt;</xsl:text>
        <xsl:text>&lt;/blockquote&gt;&#10;&amp;lt;/&lt;font color=&quot;NAVY&quot;&gt;</xsl:text>
        <xsl:value-of select="local-name()"/>
        <xsl:text>&lt;/font&gt;&amp;gt;&#10;</xsl:text>
        <xsl:text>&lt;br&gt;&#10;</xsl:text>
      </xsl:template>
      </xsl:stylesheet>
```

F. X3DTOHTRML OUTPUT HELLOWORLD.HTML

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "http://www.Web3D.org/TaskGroups/x3d/translation/x3d-compact.dtd"
                     "file:///C:/www.Web3D.org/TaskGroups/x3d/translation/x3d-compact.dtd">
<-- type="text/xsl" href="X3dHTML.xsl" --> <-- type="text/xsl" href="X3dVRML.xsl" --> <X3D
version='3.0'>

<head>

<meta name='filename' content='HelloWorld.xml'/>
<meta name='description' content='Simple X3D example'/>
<meta name='revised' content='30 October 2000'/>
<meta name='author' content='Don Brutzman'/>
```

```
<meta name='url'
content='http://www.Web3D.org/TaskGroups/x3d/translation/examples/HelloWorld.xml'/>
<meta name='generator' content='X3D-Edit,
http://www.Web3d.org/TaskGroups/x3d/translation/README.X3D-Edit.html'/>
</head>
<Scene>
<!-- Example scene to illustrate X3D tags and attributes. -->
<Group>
<Viewpoint description='XML Presentation!' orientation='0 1 0 1.57' position='9 -
1 0'/>
<NavigationInfo type='"EXAMINE" "ANY"'/>
<Shape>
<Sphere DEF='S'/>
<Appearance>
<ImageTexture url='"earth-topo.png" "earth-topo-small.gif"
"C://www.Web3D.org/TaskGroups/x3d/translation/examples/earth-topo.png"
"C://www.Web3D.org/TaskGroups/x3d/translation/examples/earth-topo-
small.gif"'/>
</Appearance>
</Shape>
<Transform rotation='0 1 0 1.57' translation='0 -2 1.25'>
<Shape>
<Text string='"XML Presentation!"'/>
<Appearance>
<Material diffuseColor='0.1 0.5 1'/>
</Appearance>
</Shape>
</Transform>
</Group>
</Scene>
```

```
</X3D>
<!-- Tag color codes: <NodeName attribute='value'/> -->
```

G. COCHELLOWORLD.XML

```
<?xml-stylesheet href="x3dvrml.xsl" type="text/xsl"?>
<!-- developed with XML Spy v4.0 U (http://www.xmlspy.com) by Clifton Williams (Naval Postgraduate School) -->

<?cocoon-process type="xslt"?>
<page>
  <title> Cocoon wrapped HelloWorld.xml  </title>
  <vrml>
#VRML V2.0 utf8
# X3D-to-VRML-97 XSL translation autogenerated by X3dToVrml97.xsl
# http://www.Web3D.org/TaskGroups/x3d/translation/X3dToVrml97.xsl

# [X3D]  VRML V3.0 utf8

# [head]
# [meta] filename: HelloWorld.xml
# [meta] description: Simple X3D example
# [meta] revised: 30 October 2000
# [meta] author: Don Brutzman
# [meta] url: http://www.Web3D.org/TaskGroups/x3d/translation/examples/HelloWorld.xml
# [meta] generator: X3D-Edit, http://www.Web3d.org/TaskGroups/x3d/translation/README.X3D-Edit.html

# [Scene]

# Example scene to illustrate X3D tags and attributes.
Group {
  children [
  Viewpoint {
    description "XML Presentation!"
    orientation 0 1 0 1.57
    position 9 -1 0
  }
  NavigationInfo {
    type [ "EXAMINE" "ANY" ]
  }
  Shape {
    geometry DEF S Sphere {
    }
    appearance Appearance {
      texture ImageTexture {
        url [ "earth-topo.png"
"earth-topo-small.gif"
"C://www.Web3D.org/TaskGroups/x3d/translation/examples/earth-topo.png"
"C://www.Web3D.org/TaskGroups/x3d/translation/examples/earth-topo-small.gif" ]
      }
    }
  }
  Transform {
    rotation 0 1 0 1.57
    translation 0 -2 1.25
    children [
    Shape {
      geometry Text {
        string [ "XML Presentation!" ]
      }
      appearance Appearance {
        material Material {
          diffuseColor 0.1 0.5 1
        }
      }
    }
    ]
  }
  ]
```

```
}
</vrml>
</page>
```

H. COCKELPFOREST.XML

```
<?xml version="1.0"?>
<?xml-stylesheet href="x3dvrml.xsl" type="text/xsl"?>
<?cocoon-process type="xslt"?>
<!-- developed with XML Spy v4.0 U (http://www.xmlspy.com) by Clifton Williams (Naval Postgraduate School) -->

<page>
  <title>Cocoon wrapped Kelp Forest</title>

#VRML V2.0 utf8
<vrml>
WorldInfo {
   title "NPS Virtual Kelp Forest Exhibit"
   info ["kelpForestMain3.4.wrl",
         "Model of the Monterey Bay Aquarium Kelp Forest",
          "DTG of last updated:  111500Jun98",
          "Added: "
          "Updated: kelpTank2.9.wrl; "
          "Created by Naval Postgraduate School students under the supervision of Don Brutzman",
          "Feel free to copy anything you wish - just give credit where it is due"
          "http://localhost/cocoon/samples/kelpForest/"
   ]
}

DEF Root Group {
   children[

      NavigationInfo {
```

```
   type ["EXAMINE" "FLY" "WALK" "ANY"]
   speed 3.0
}

Viewpoint {
   description "Kelp Forest Monterey Bay Aquarium "
   position  0.0 -5.0 45.0
}

Viewpoint {
   description "Front Windows"
   position 0.0 -4.5 7.5
   orientation 1 0 0 .15
   fieldOfView .95
}

Viewpoint {
   description "Side Windows"
   position 25 -4 15
   orientation 0 1 0 .9
   fieldOfView .5
}

Viewpoint {
   description "Star Fish"
   position 14 -5.75 1
   orientation 0 1 0 1.25
}

Viewpoint {
   description "Inside Tank"
   position 0.0 -5.0 -7.0
   orientation 0.0 1.0 0.0 3.14
}
```

```
Viewpoint {
   description "Top floor"
   position 12.5 5 -2
   orientation -0.3 1 0.3 1.57079
   fieldOfView  1.04719
}

Viewpoint {
   description "Bird's Eye View"
   position 0.0 19.0 -3.5
   orientation 1.0 0.0 0.0 -1.57
}

Viewpoint {
   description "Pump House"
   position -5 1 0
   orientation 0.0 1.0 0.0 1.3
   fieldOfView .6
}

Background {
   skyColor [
      0.0 0.5 1.0,
      0.0 0.0 1.0,
      0.0 0.0 1.0
   ]
   skyAngle [ 1.309, 1.571 ]
   groundColor [
      0.1 0.10 0.0,
      0.4 0.25 0.2,
      0.6 0.60 0.6,
   ]
   groundAngle [ 1.309, 1.571 ]
```

```
      }

      Inline {
         url                             [                              "file:///C:/jakarta-tomcat-
3.2.3/Webapps/cocoon/samples/vrml/kelpForest/introductionMessage1.3.wrl"
         "introductionMessage1.3.wrl"
]
      }

      Collision {
         collide FALSE
         children [
            Transform{
               children [
                  Inline {
                     url["file:///C:/jakarta-tomcat-
3.2.3/Webapps/cocoon/samples/vrml/kelpForest/sharkLocaleScott5.wrl"
                     "sharkLocaleScott5.wrl" #calls shark2.2.wrl
                   ]
                  }
               ]
            }
            Transform{
               children [
                  Inline {
                     url["file:///C:/jakarta-tomcat-
3.2.3/Webapps/cocoon/samples/vrml/kelpForest/sharkLocaleScott4.wrl"
                      "sharkLocaleScott4.wrl" #calls shark2.1.wrl
                   ]
                  }
               ]
            }

            Inline {
               url["file:///C:/jakarta-tomcat-
3.2.3/Webapps/cocoon/samples/vrml/kelpForest/kelpTank2.9.wrl"
                "kelpTank2.9.wrl"
                 ]
            }

            Transform{
               translation 0.3 -3.3 7.3
               scale 0.54 0.75 0.52
               children[
                  Inline{
                     url["file:///C:/jakarta-tomcat-
3.2.3/Webapps/cocoon/samples/vrml/kelpForest/rockFloor1.5.wrl"
                     "rockFloor1.5.wrl"
                    ]
                  }
               ]
            }

#       DEF K1_LOCALE Transform {
#            translation  -10.0 0.0 2.0
#            children [

#               DEF K1_SHALLOW Tansform {
#                  translation 0.0 0.0 0.0
#                  children [
#                  ]
#                }#end K1_SHALLOW

#               DEF K1_MIDWATER Transform {
#                  translation 0.0 0.0 0.0
#                  children [
#                  ]
#              }#end K1_MIDWATER
#               DEF K1_DEEP Transform {
#                  translation 0.0 0.0 0.0
#                  children [
#                   ]
#               }#end K1_DEEP

#               DEF K1_FLOOR Transform {
#                  translation 0.0 0.0 0.0
#                  children [
#                  ]
#              }
#            ]
#         }#end K1_LOCALE Transform


         DEF K2_LOCALE Transform {
            translation  -4 0 1
            children [

#               DEF K2_SHALLOW Tansform {
#                  translation 0.0 0.0 0.0
#                  children [
#                  ]
#                }#end K2_SHALLOW

               DEF K2_MIDWATER Transform {
                  translation -1 -4 -4
                  scale .20 .20 .20
                  children [
                     Inline {
url [  "file:///C:/jakarta-tomcat-
3.2.3/Webapps/cocoon/samples/vrml/kelpForest/SardineSchool.gz.wrl"
```

```
  "file:///C:/jakarta-tomcat-
3.2.3/Webapps/cocoon/samples/vrml/kelpForest/SardineSchool/SardineSchool.gz.wrl"
  "file:///C:/jakarta-tomcat-3.2.3/Webapps/cocoon/samples/vrml/kelpForest/SardineSchool.wrl"
  "file:///C:/jakarta-tomcat-
3.2.3/Webapps/cocoon/samples/vrml/kelpForest/SardineSchool/SardineSchool.wrl"
     "SardineSchool.gz.wrl"
  "SardineSchool/SardineSchool.gz.wrl"
  "SardineSchool.wrl"
  "SardineSchool/SardineSchool.wrl"
  # modeled by Stewart Liles
    ]
                     }
                 ]
               }#end K2_MIDWATER

#               DEF K2_DEEP Transform {
#                  translation 0.0 0.0 0.0
#                  children [
#                   ]
#               }#end K2_DEEP

               DEF K2_FLOOR Transform {
                  translation -1.9 -8.5 -1.5
                  rotation 0 1 0 .44
                  scale .5 .9 .5
                  children [
                    Inline {
                       url                                            ["file:///C:/jakarta-tomcat-
3.2.3/Webapps/cocoon/samples/vrml/kelpForest/kelpNoBase.wrl"
                       "kelpNoBase.wrl"
                      ]
                    }
                  ]
               }
```

```
            ]
         }#end K2_LOCALE Transform

#      DEF K3_LOCALE Transform {
#         translation  -10.0 0.0 2.0
#         children [

#            DEF K3_SHALLOW Tansform {
#               translation 0.0 0.0 0.0
#               children [
#                ]
#             }#end K3_SHALLOW

#            DEF K3_MIDWATER Transform {
#               translation 0.0 0.0 0.0
#               children [
#               ]
#            }#end K3_MIDWATER

#            DEF K3_DEEP Transform {
#               translation 0.0 0.0 0.0
#               children [
#                ]
#             }#end K3_DEEP

#            DEF K3_FLOOR Transform {
#               translation 0.0 0.0 0.0
#               children [
#               ]
#            }
#         ]
#      }#end K3_LOCALE Transform
```

```
      DEF K4_LOCALE Transform {
         translation  -2.5 0 -2.5
         children [

#            DEF K4_SHALLOW Tansform {
#               translation 0.0 0.0 0.0
#               children [
#                ]
#             }#end K4_SHALLOW

            DEF K4_MIDWATER Transform {
               translation -10.0 -2.0 -2
               children [
                  Inline {
                     url["file:///C:/jakarta-tomcat-
3.2.3/Webapps/cocoon/samples/vrml/kelpForest/HalfmoonGroup.wrl"
                     "HalfmoonGroup.wrl"
                     ]
                  }
               ]
            }#end K4_MIDWATER

#            DEF K4_DEEP Transform {
#               translation 0.0 0.0 0.0
#               children [
#                ]
#             }#end K4_DEEP

            DEF K4_FLOOR Transform {
               translation -1.5 -7.5 .5
               rotation 0 1 0 -.3
               scale .5 .8 .5
               children [
                  Inline {
                       url["file:///C:/jakarta-tomcat-
3.2.3/Webapps/cocoon/samples/vrml/kelpForest/kelpNoBase.wrl"
                       "kelpNoBase.wrl"
          ]
                  }
               ]
            }
         ]
      }#end K4_LOCALE Transform

#      DEF K5_LOCALE Transform {
#         translation  -10.0 0.0 2.0
#         children [

#            DEF K5_SHALLOW Tansform {
#               translation 0.0 0.0 0.0
#               children [
#                ]
#             }#end K5_SHALLOW

#            DEF K5_MIDWATER Transform {
#               translation 0.0 0.0 0.0
#               children [
#               ]
#            }#end K5_MIDWATER

#            DEF K5_DEEP Transform {
#               translation 0.0 0.0 0.0
#               children [
#                ]
#             }#end K5_DEEP

#            DEF K5_FLOOR Transform {
#               translation 0.0 0.0 0.0
#               children [
#               ]
#            }
#         ]
#      }#end K5_LOCALE Transform

#      DEF K6_LOCALE Transform {
#         translation  -10.0 0.0 2.0
#         children [

#            DEF K6_SHALLOW Tansform {
#               translation 0.0 0.0 0.0
#               children [
#                ]
#             }#end K6_SHALLOW

#            DEF K6_MIDWATER Transform {
#               translation 0.0 0.0 0.0
#               children [
#               ]
#            }#end K6_MIDWATER

#            DEF K6_DEEP Transform {
#               translation 0.0 0.0 0.0
#               children [
#                ]
#             }#end K6_DEEP

#            DEF K6_FLOOR Transform {
#               translation 0.0 0.0 0.0
#               children [
#               ]
#            }
#         ]
#      }#end K6_LOCALE Transform

      DEF K7_LOCALE Transform {
         translation  2 0 -2.5
         children [

#            DEF K7_SHALLOW Tansform {
#               translation 0.0 0.0 0.0
#               children [
#                ]
#             }#end K7_SHALLOW

           DEF K7_MIDWATER Transform {
              translation 2.0 -4.0 0.0
              children [
                 Inline {
                    url["file:///C:/jakarta-tomcat-
3.2.3/Webapps/cocoon/samples/vrml/kelpForest/rubberlip.wrl"
                    "rubberlip.wrl"
                  ]
                 }
              ]
              scale 0.25 0.25 0.25
           }#end K7_MIDWATER

#            DEF K7_DEEP Transform {
#               translation 0.0 0.0 0.0
#               children [
#                ]
#             }#end K7_DEEP

            DEF K7_FLOOR Transform {
               translation 0 -7.75 0
               scale .5 .85 .5
               children [
                 Inline {
                    url["file:///C:/jakarta-tomcat-
3.2.3/Webapps/cocoon/samples/vrml/kelpForest/kelpNoBase.wrl"
                    "kelpNoBase.wrl"
                ]
                 }
               ]
            }
         ]
      }#end K7_LOCALE Transform

#      DEF K8_LOCALE Transform {
#         translation  -10.0 0.0 2.0
#         children [

#            DEF K8_SHALLOW Tansform {
#               translation 0.0 0.0 0.0
#               children [
#                ]
#             }#end K8_SHALLOW

#            DEF K8_MIDWATER Transform {
#               translation 0.0 0.0 0.0
#               children [
#               ]
#            }#end K8_MIDWATER

#            DEF K8_DEEP Transform {
#               translation 0.0 0.0 0.0
#               children [
#                ]
#             }#end K8_DEEP
#            DEF K8_FLOOR Transform {
#               translation 0.0 0.0 0.0
#               children [
#               ]
#            }
#         ]
#      }#end K8_LOCALE Transform

#      DEF K9_LOCALE Transform {
#         translation  -10.0 0.0 2.0
#         children [

#            DEF K9_SHALLOW Tansform {
#               translation 0.0 0.0 0.0
#               children [
#                ]
#             }#end K9_SHALLOW

#            DEF K9_MIDWATER Transform {
#               translation 0.0 0.0 0.0
#               children [
#               ]
#            }#end K9_MIDWATER

#            DEF K9_DEEP Transform {
#               translation 0.0 0.0 0.0
#               children [
#                ]
#             }#end K9_DEEP

#            DEF K9_FLOOR Transform {
#               translation 0.0 0.0 0.0
#               children [
#               ]
#            }
#          ]
#       }#end K9_LOCALE Transform

      DEF K10_LOCALE Transform {
         translation  7.0 0.0 2.0
         children [

#            DEF K10_SHALLOW Tansform {
#               translation 0.0 0.0 0.0
#               children [
#                ]
#             }#end K10_SHALLOW

#            DEF K10_MIDWATER Transform {
#               translation 0.0 0.0 0.0
#               children [
#               ]
#            }#end K10_MIDWATER

            DEF K10_DEEP Transform {
               translation 0 -5 0
               children [
                 Inline {
                    url["file:///C:/jakarta-tomcat-
3.2.3/Webapps/cocoon/samples/vrml/kelpForest/StripedSurfperch.wrl"
                     "StripedSurfperch.wrl"
         ]
                 }
               ]
            }#end K10_DEEP

            DEF K10_FLOOR Transform {
               translation -1 -7 -1
               rotation 0.0 1.0 0.0 .16
               scale .5 .75 .5
               children [
                  Inline {
                     url[“file:///C:/jakarta-tomcat-
3.2.3/Webapps/cocoon/samples/vrml/kelpForest/kelpNoBase.wrl"
                     "kelpNoBase.wrl"
                       ]
                  }
               ]
            }
         ]
      }#end K10_LOCALE Transform

      DEF K11_LOCALE Transform {
         translation  8 0 -2
         children [

#            DEF K11_SHALLOW Tansform {
#               translation 0.0 0.0 0.0
#               children [
#                ]
#             }#end K11_SHALLOW

            DEF K11_MIDWATER Transform {
                translation 2.19 -7.0 0.15
                rotation 0 1 0 .3
                children [
                   Inline {
                      url["file:///C:/jakarta-tomcat-
3.2.3/Webapps/cocoon/samples/vrml/kelpForest/starFish1.8.wrl"
                      "starFish1.8.wrl"
                    ]
                   }
                ]
            }#end K11_MIDWATER

            DEF K11_DEEP Transform {
               translation 1.0 -5 0
               rotation 0 1 0 .9
               children [
                  Inline {
                     url["file:///C:/jakarta-tomcat-
3.2.3/Webapps/cocoon/samples/vrml/kelpForest/BlackSurfperch.wrl"
                     "BlackSurfperch.wrl" #moves in and out of rock
                      ]
                  }
                ]
             }#end K11_DEEP

            DEF K11_FLOOR Transform {
               translation -.3 -7 0
               rotation 0 1 0 .3
               scale .5 .75 .5
               children [
                  Inline {
                     url["file:///C:/jakarta-tomcat-
3.2.3/Webapps/cocoon/samples/vrml/kelpForest/kelpNoBase.wrl"
                     "kelpNoBase.wrl"
              ]
                  }
               ]
            }
            Transform {
                  translation 1.3 -5.0 3.0
                  children [
                     Inline {
                        url["file:///C:/jakarta-tomcat-
3.2.3/Webapps/cocoon/samples/vrml/kelpForest/starFishPatch.wrl"
                         "starFishPatch.wrl"
                   ]
                     }
                  ]
               }
         ]
      }#end K11_LOCALE Transform

#       DEF R1_LOCALE Transform {
#            translation  -10.0 0.0 2.0
#            children [

#               DEF R1_SHALLOW Tansform {
#                  translation 0.0 0.0 0.0
#                  children [
#                  ]
#                }#end R1_SHALLOW

#               DEF R1_MIDWATER Transform {
#                  translation 0.0 0.0 0.0
#                  children [
#                  ]
#              }#end R1_MIDWATER

#               DEF R1_DEEP Transform {
#                  translation 0.0 0.0 0.0
#                  children [
#                   ]
#               }#end R1_DEEP

#               DEF R1_FLOOR Transform {
#                  translation 0.0 0.0 0.0
#                  children [
#                  ]
#              }
#            ]
#         }#end R1_LOCALE Transform

        DEF R2_LOCALE Transform {
          translation  -1 0  -1
               children [

#               DEF R2_SHALLOW Tansform {
#                  translation 0.0 0.0 0.0
#                  children [
#                  ]
#                }#end R2_SHALLOW

             DEF R2_MIDWATER Transform {
                   translation -1 -6 -2
                   children [
                     Inline {
                        url["file:///C:/jakarta-tomcat-
3.2.3/Webapps/cocoon/samples/vrml/kelpForest/blueRockFish.wrl"
                        "blueRockFish.wrl"
                     ]
                     }

                   ]
               }#end R2_MIDWATER

                DEF R2_DEEP Transform {
                   translation 1 -6.5 -1
                   children [
                      Inline {
                        url["file:///C:/jakarta-tomcat-
3.2.3/Webapps/cocoon/samples/vrml/kelpForest/kelpSurfperch.wrl"
                        "kelpSurfperch.wrl"
                 ]
                     }


                    ]
                }#end R2_DEEP

#               DEF R2_FLOOR Transform {
#                  translation 0.0 0.0 0.0
#                  children [
#                  ]
#              }
             ]
          }#end R2_LOCALE Transform

        DEF R3_LOCALE Transform {
             translation  3 0 -4
             children [

#               DEF R3_SHALLOW Tansform {
#                  translation 0.0 0.0 0.0
#                  children [
#                  ]
#                }#end R3_SHALLOW

                DEF R3_MIDWATER Transform {
                   translation 0.0 -6 0.0
                   children [
                     Inline {
                        url["file:///C:/jakarta-tomcat-
3.2.3/Webapps/cocoon/samples/vrml/kelpForest/garibaldi.wrl"
```

```
                        "garibaldi.wrl"
                        ]
                     }

                   ]
               }#end R3_MIDWATER

#               DEF R3_DEEP Transform {
#                  translation 0.0 0.0 0.0
#                  children [
#                   ]
#               }#end R3_DEEP

#               DEF R3_FLOOR Transform {
#                  translation 0.0 0.0 0.0
#                  children [
#                  ]
#              }
             ]
          }#end R3_LOCALE Transform

#        DEF R4_LOCALE Transform {
#             translation  7 0.0 2.0
#             children [

#               DEF R4_SHALLOW Tansform {
#                  translation 0.0 0.0 0.0
#                  children [
#                  ]
#                }#end R4_SHALLOW

#               DEF R4_MIDWATER Transform {
#                  translation 0.0 0.0 0.0
#                  children [
#                  ]
#              }#end R4_MIDWATER

#               DEF R4_DEEP Transform {
#                  translation 0.0 0.0 0.0
#                  children [
#                   ]
#               }#end R4_DEEP

#               DEF R4_FLOOR Transform {
#                  translation 0.0 0.0 0.0
#                  children [
#                  ]
#              }
#            ]
#         }#end R4_LOCALE Transform

        DEF R5_LOCALE Transform {
             translation  8 0 -2
             children [

                DEF R5_SHALLOW Transform {
                   translation -2 -2 0
                   children [
                     Inline {
                        url["file:///C:/jakarta-tomcat-
3.2.3/Webapps/cocoon/samples/vrml/kelpForest/kelpBass.wrl"
                         "kelpBass.wrl"
                     ]
                     }

                   ]
                 }#end R5_SHALLOW
```

```
#               DEF R5_MIDWATER Transform {
#                  translation 0.0 0.0 0.0
#                  children [
#                  ]
#              }#end R5_MIDWATER

#               DEF R5_DEEP Transform {
#                  translation 0.0 0.0 0.0
#                  children [
#                   ]
#               }#end R5_DEEP

#               DEF R5_FLOOR Transform {
#                  translation 0.0 0.0 0.0
#                  children [
#                  ]
#              }
             ]
          }#end R5_LOCALE Transform

      DEF WAVE_CLOCK TimeSensor {
         cycleInterval 5.5
         enabled FALSE # master startup clock will initialize
all TimeSensors synchronously
         loop TRUE
      }

      DEF LEFT_SHALLOW_WAVE PositionInterpolator{
         key[0, .125, .25, .375, .5, .625, .75, .875, 1
         ]
         keyValue[
             0       0      0,
             0.056  -0.085  0,
             0.08   -0.12   0,
       0.056  -0.085  0,
       0       0      0,
      -0.056   0.085  0,
      -0.08     0.12   0,
      -0.056   0.085  0,
       0       0      0
   ]
}
DEF LEFT_DEEP_WAVE PositionInterpolator{
   key[0, .125, .25, .375, .5, .625, .75, .875, 1
   ]
   keyValue[
       0       0      0,
       0.035  -0.035  0,
       0.05   -0.05   0,
       0.035  -0.035  0,
       0       0      0,
      -0.035   0.035  0,
      -0.05    0.05   0,
      -0.035   0.035  0,
       0       0      0
   ]
}
DEF CENTER_WAVE PositionInterpolator{
   key[0, .125, .25, .375, .5, .625, .75, .875, 1
   ]
   keyValue[
       0       0      0,
       0.106   0      0,
       0.15    0      0,
       0.106   0      0,
       0       0      0,
      -0.106   0      0
      -0.15    0      0,
      -0.106   0      0,
       0       0      0
   ]
}
DEF RIGHT_SHALLOW_WAVE PositionInterpolator{
   key[0, .125, .25, .375, .5, .625, .75, .875, 1
   ]
   keyValue[
       0       0      0,
       0.056   0.085  0,
       0.08    0.12   0,
       0.056   0.085  0,
       0       0      0,
      -0.056  -0.085  0,
      -0.08   -0.12   0,
      -0.056  -0.085  0,
       0       0      0
   ]
}

DEF RIGHT_DEEP_WAVE PositionInterpolator{
   key[0, .125, .25, .375, .5, .625, .75, .875, 1
   ]
   keyValue[
       0       0      0,
       0.035   0.035  0,
       0.05    0.05   0,
       0.035   0.035  0,
       0       0      0,
      -0.035  -0.035  0,
      -0.05   -0.05   0,
      -0.035  -0.035  0,
       0       0      0
   ]
```

```
      }


   ]#end collision children
}#end collision node

   ]
}#end kelpForestMain

#ROUTE WAVE_CLOCK.fraction_changed TO LEFT_SHALLOW_WAVE.set_fraction
#ROUTE WAVE_CLOCK.fraction_changed TO LEFT_DEEP_WAVE.set_fraction
ROUTE WAVE_CLOCK.fraction_changed TO CENTER_WAVE.set_fraction
#ROUTE WAVE_CLOCK.fraction_changed TO RIGHT_SHALLOW_WAVE.set_fraction
#ROUTE WAVE_CLOCK.fraction_changed TO RIGHT_DEEP_WAVE.set_fraction
# also need to route this clock to pump and water surface,
# so put all of these sensors/routes/transforms in top-level scene

ROUTE CENTER_WAVE.value_changed  TO  K7_MIDWATER.set_translation
ROUTE RIGHT_DEEP_WAVE.value_changed   TO  K2_MIDWATER.set_translation
ROUTE CENTER_WAVE.value_changed  TO  K4_MIDWATER.set_translation
ROUTE LEFT_DEEP_WAVE.value_changed    TO  K10_DEEP.set_translation
ROUTE LEFT_DEEP_WAVE.value_changed    TO  K11_DEEP.set_translation
ROUTE RIGHT_SHALLOW_WAVE.value_changed  TO  R2_MIDWATER.set_translation
ROUTE RIGHT_DEEP_WAVE.value_changed     TO  R2_DEEP.set_translation
ROUTE CENTER_WAVE.value_changed  TO  R3_MIDWATER.set_translation
ROUTE RIGHT_SHALLOW_WAVE.value_changed  TO  R5_SHALLOW.set_translation
# etc. route one of five wave clocks to each locale
   </vrml>
</page>
```

I. COCAMPHIBIOUSRAID.XML

```
<?xml version="1.0"?>
<!-- developed with XML Spy v4.0 U (http://www.xmlspy.com) by Clifton Williams (Naval Postgraduate School) -->
<?xml-stylesheet href="x3dvrml.xsl" type="text/xsl"?>
<?cocoon-process type="xslt"?>
<!-- Edited by Clifton Williams" -->
<page>
 <title> Cocoon wrapped AmphibiousRaid.xml </title>

#VRML V2.0 utf8
<vrml>
    <!-- This is the start of the WRL file code -->


# X3D-to-VRML-97 XSL translation autogenerated by X3dToVrml97.xsl
# http://www.web3D.org/TaskGroups/x3d/translation/X3dToVrml97.xsl

# [X3D]  VRML V3.0 utf8

# [head]
# [meta] filename: AmphibiousRaid.xml
# [meta] description: Amphibious Raid shows an exemplar amphibious raid at Red Beach, Camp Pendleton California.  It is the primary scenario for the SAVAGE group.
# [meta] author: Don Brutzman, Curtis Blais, Jeff Weekley, Jane Wu, Shane Nicklaus, MIke Hunsberger
# [meta] created: 9 April 2001
# [meta] revised: 27 July 2001
# [meta] reference: NpsRedBeachRaidScenario2001May2.html
# [meta] reference: NpsRedBeachRaidScenario2001May2.ppt
# [meta] image: HeloPatrolPendletonBackground.png
# [meta] reference: AmphibiousRaidWithCommunications.xml
# [meta] warning: Coordinate system axes need to be aligned properly.
```

```
     # [meta] url:
http://web.nps.navy.mil/~brutzman/Savage/Scenarios/CampPendletonCalifornia/AmphibiousRaid.xml
     # [meta] reference:
http://web.nps.navy.mil/~brutzman/Savage/Tools/Animation/WaypointInterpolator.xml
     # [meta] generator: X3D-Edit, http://www.web3D.org/TaskGroups/x3d/translation/README.X3D-
Edit.html

     EXTERNPROTO EspduTransform [
      eventOut  SFTime   timestamp# DIS timestamp in VRML units
      field     SFString marking    # 0..11 character label for entity
      field     SFTime   readInterval  # seconds between read  updates, 0 means no reading
      field     SFTime   writeInterval # seconds between write updates, 0 means no writing
      field     SFString address    # multicast address or "localhost"
      field     SFInt32    port   # port number
      field     SFInt32    siteID   # EntityID triplet, first element
      field     SFInt32    applicationID # EntityID triplet, second element
      field     SFInt32    entityID # EntityID triplet, third element

      # Standard VRML Transform fields - - - - - - - - - - - - - - - - - - - - - - - - -
      exposedField  MFNode   children
      exposedField  SFVec3f    translation     # default prior to DIS update
      exposedField  SFRotation rotation        # default prior to DIS update
      exposedField  SFVec3f    scale
      exposedField  SFRotation scaleOrientation
      field     SFVec3f    bboxCenter
      field     SFVec3f    bboxSize
      exposedField  SFVec3f    center
      eventIn   MFNode   addChildren
      eventIn   MFNode   removeChildren

      # Automatic MulticastRelayClient if no multicast heard - - - - - - - - - - - - - -
      field     SFString multicastRelayHost
      field     SFInt32  multicastRelayPort
      field     SFBool   rtpHeaderExpected # whether RTP headers are prepended to DIS
      eventOut  SFBool   rtpHeaderHeard

      # Collision/Detonation/Fire PDU heard, or no active ESPDUs heard - - - - - - - - -
      eventOut  SFBool   active   # have we had an update recently?
      eventOut  SFBool   collided # has a CollisionPDU reported a collision?
      eventOut  SFTime   collideTime # when were we collided with?
      eventOut  SFBool   detonated# has a DetonationPDU reported a detonation?
      eventOut  SFTime   detonateTime  # when were we detonated?
      eventOut  SFBool   fired1   # have we shot a Fire (weapon) PDU (primary)?
      eventOut  SFBool   fired2   # have we shot a Fire (weapon) PDU (secondary)?
      eventOut  SFTime   firedTime# when did we shoot a Fire (weapon) PDU?
      eventOut  SFVec3f    munitionStartPoint # exercise coordinates
      eventOut  SFVec3f    munitionEndPoint # exercise coordinates

      # Articulated Parameter (user-defined payload) fields  - - - - - - - - - - - - - - - - - - -
- - - -
      eventOut  SFInt32    articulationParameterCount# articulated parameters available
      eventOut  SFFloat    articulationParameterValue0# user-defined payload value
      eventOut  SFFloat    articulationParameterValue1# user-defined payload value
      eventOut  SFFloat    articulationParameterValue2# user-defined payload value
      eventOut  SFFloat    articulationParameterValue3# user-defined payload value
      eventOut  SFFloat    articulationParameterValue4# user-defined payload value
      eventOut  SFFloat    articulationParameterValue5# user-defined payload value
      eventOut  SFFloat    articulationParameterValue6# user-defined payload value
      eventOut  SFFloat    articulationParameterValue7# user-defined payload value
      eventOut  SFFloat    articulationParameterValue8# user-defined payload value
      eventOut  SFFloat    articulationParameterValue9# user-defined payload value
      eventOut  SFFloat    articulationParameterValue10# user-defined payload value
      eventOut  SFFloat    articulationParameterValue11# user-defined payload value
      eventOut  SFFloat    articulationParameterValue12# user-defined payload value
      eventOut  SFFloat    articulationParameterValue13# user-defined payload value
      eventOut  SFFloat    articulationParameterValue14# user-defined payload value

      # Trace fields - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      exposedField  SFColor    traceColor # color of VRML trace text
      exposedField  SFVec3f    traceOffset # VRML text position relative to entity
      exposedField  SFVec3f    traceSize# VRML text scale
      field    SFBool   traceJava# Java console tracing enabled?

      ] [#"EspduTransformPROTO.wrl#EspduTransform"# local or remote URLs for the EXTERNPROTO
       "file:///C:/vrtp/mil/navy/nps/dis/EspduTransformPROTO.wrl#EspduTransform"
       "file:///D:/vrtp/mil/navy/nps/dis/EspduTransformPROTO.wrl#EspduTransform"
       "file:///C|/vrtp/mil/navy/nps/dis/EspduTransformPROTO.wrl#EspduTransform"
       "file:///D|/vrtp/mil/navy/nps/dis/EspduTransformPROTO.wrl#EspduTransform"
       "file:///vrtp/mil/navy/nps/dis/EspduTransformPROTO.wrl#EspduTransform"
       "file://vrtp/mil/navy/nps/dis/EspduTransformPROTO.wrl#EspduTransform"

"http://web.nps.navy.mil/~brutzman/vrtp/mil/navy/nps/dis/EspduTransformPROTO.wrl#EspduTransform"

"http://www.web3D.org/WorkingGroups/vrtp/mil/navy/nps/dis/EspduTransformPROTO.wrl#EspduTransform"

       "/vrtp/mil/navy/nps/dis/EspduTransformPROTO.wrl#EspduTransform"
       "../vrtp/mil/navy/nps/dis/EspduTransformPROTO.wrl#EspduTransform"
       "../../mil/navy/nps/dis/EspduTransformPROTO.wrl#EspduTransform"
       "../../vrtp/mil/navy/nps/dis/EspduTransformPROTO.wrl#EspduTransform"
       "../../../vrtp/mil/navy/nps/dis/EspduTransformPROTO.wrl#EspduTransform"
       "../../../../vrtp/mil/navy/nps/dis/EspduTransformPROTO.wrl#EspduTransform"
       "../../../../../vrtp/mil/navy/nps/dis/EspduTransformPROTO.wrl#EspduTransform"
       "../../../../../../vrtp/mil/navy/nps/dis/EspduTransformPROTO.wrl#EspduTransform"
       "../../../../../../../vrtp/mil/navy/nps/dis/EspduTransformPROTO.wrl#EspduTransform"
       "../../../../../../../../vrtp/mil/navy/nps/dis/EspduTransformPROTO.wrl#EspduTransform"
       "../../../../../../../../../vrtp/mil/navy/nps/dis/EspduTransformPROTO.wrl#EspduTransform"

"../../../../../../../../../../vrtp/mil/navy/nps/dis/EspduTransformPROTO.wrl#EspduTransform"
        ]
     # [Scene]

     # Prototype and external prototype declarations must precede scene definition
```

```
      EXTERNPROTO HiddenViewpoint [
        exposedField SFVec3f  position
        exposedField SFRotation rotation
        field        SFFloat  sensorRadius
        exposedField MFString label
        exposedField SFVec3f  labelOffset
        field        SFFloat  labelFontSize
        exposedField SFColor  labelColor
      ][
        "file:///C:/jakarta-tomcat-
3.2.3/webapps/cocoon/samples/vrml/Savage/Tools/Animation/HiddenViewpointPrototype.wrl#HiddenViewpo
int"
        "../../Tools/Animation/HiddenViewpointPrototype.wrl#HiddenViewpoint"
      "http://web.nps.navy.mil/~brutzman/Savage/Tools/Animation/HiddenViewpointPrototype.wrl#Hidde
nViewpoint"
      "file:///C:/jakarta-tomcat-
3.2.3/webapps/cocoon/samples/vrml/Savage/Tools/Animation/HiddenViewpointPrototype.xml#HiddenViewpo
int"
      "../../Tools/Animation/HiddenViewpointPrototype.xml#HiddenViewpoint"
      "http://web.nps.navy.mil/~brutzman/Savage/Tools/Animation/HiddenViewpointPrototype.xml#Hidde
nViewpoint"
       ]
      EXTERNPROTO WaypointInterpolator [
      # nodeTypeHint Group
        field        MFVec3f  wayPoints
        field        SFBool pitchUpDownForVerticalWayPoints
        # Priority of use: legSpeeds (m/sec), legDurations (seconds), defaultSpeed (m/sec)
        # If used, array lengths for legSpeeds and legDurations must be one less than number of
wayPoints.
        field        MFFloat  legSpeeds
        field        MFTime legDurations
        field        SFFloat  defaultSpeed
        # turningRate (degrees/second) will also determine standoff distance prior to waypoint
where turn commences. If 0, turns are instantaneous.
```

```
        field        SFFloat  turningRate
        eventOut     SFTime totalDuration
        # exposed PositionInterpolator and OrientationInterpolator settings:
        eventIn      SFFloat  fraction
        eventOut     SFVec3f  positionValue
        eventOut     SFRotation orientationValue
        exposedField SFColor  lineColor
        # highlightSegmentColors must contain two color values, for each endpoint of the highlight
segment.
        exposedField MFColor  highlightSegmentColors ### XSL-translation warning:  IS declarations
not allowed for Script or ExternProtoDeclare, ignoring IS value
       ###  XSL-translation  warning:   IS  references  not  allowed  inside  EXTERNPROTO  field
declarations, ignored ### IS HighlightSegmentColorNode.color
        exposedField SFFloat  transparency
        field        SFVec3f  labelOffset
        field        SFFloat  labelFontSize
        field        SFColor  labelColor
        field        SFBool traceEnabled
      ][
        "file:///C:/jakarta-tomcat-
3.2.3/webapps/cocoon/samples/vrml/Savage/Tools/Animation/WaypointInterpolatorPrototype.wrl#Waypoin
tInterpolator"
        "../../Tools/Animation/WaypointInterpolatorPrototype.wrl#WaypointInterpolator"
      "http://web.nps.navy.mil/~brutzman/Savage/Tools/Animation/WaypointInterpolatorPrototype.wrl#
WaypointInterpolator"

      "file:///C:/jakarta-tomcat-
3.2.3/webapps/cocoon/samples/vrml/Savage/Tools/Animation/WaypointInterpolatorPrototype.xml#Waypoin
tInterpolator"
      "../../Tools/Animation/WaypointInterpolatorPrototype.xml#WaypointInterpolator"
      "http://web.nps.navy.mil/~brutzman/Savage/Tools/Animation/WaypointInterpolatorPrototype.xml#
WaypointInterpolator"

       ]
      EXTERNPROTO ViewPositionOrientation [
        exposedField SFBool enabled
      ][
        "file:///C:/jakarta-tomcat-
3.2.3/webapps/cocoon/samples/vrml/Savage/Tools/Authoring/ViewPositionOrientationPrototype.wrl#View
PositionOrientation"
        "../../Tools/Authoring/ViewPositionOrientationPrototype.wrl#ViewPositionOrientation"
      "http://web.nps.navy.mil/~brutzman/Savage/Tools/Authoring/ViewPositionOrientationPrototype.w
rl#ViewPositionOrientation"
      "file:///C:/jakarta-tomcat-
3.2.3/webapps/cocoon/samples/vrml/Savage/Tools/Authoring/ViewPositionOrientationPrototype.xml#View
PositionOrientation"
      "../../Tools/Authoring/ViewPositionOrientationPrototype.xml#ViewPositionOrientation"
      "http://web.nps.navy.mil/~brutzman/Savage/Tools/Authoring/ViewPositionOrientationPrototype.x
ml#ViewPositionOrientation"
       ]
      EXTERNPROTO CameraCompass36 [
        exposedField SFBool enabled
        exposedField SFVec3f  positionOffsetFromCamera
        exposedField SFColor  markerColor
        exposedField SFColor  labelColor
      ][
        "file:///C:/jakarta-tomcat-
3.2.3/webapps/cocoon/samples/vrml/Savage/Tools/HeadsUpDisplays/CameraCompassPrototypes.wrl#CameraC
ompass36"
        "../../Tools/HeadsUpDisplays/CameraCompassPrototypes.wrl#CameraCompass36"
      "http://web.nps.navy.mil/~brutzman/Savage/Tools/HeadsUpDisplays/CameraCompassPrototypes.wrl#
CameraCompass36"
      "file:///C:/jakarta-tomcat-
3.2.3/webapps/cocoon/samples/vrml/Savage/Tools/HeadsUpDisplays/CameraCompassPrototypes.xml#CameraC
ompass36"
      "../../Tools/HeadsUpDisplays/CameraCompassPrototypes.xml#CameraCompass36"
      "http://web.nps.navy.mil/~brutzman/Savage/Tools/HeadsUpDisplays/CameraCompassPrototypes.xml#
CameraCompass36"
```

```
      ]
     EXTERNPROTO AH1SuperCobra [
     # nodeTypeHint Group
       field        SFNode     tailBoomColor
       field        SFNode     tailRotorDriveShaftColor
       field        SFNode     horizontalStabilizerColor
       field        SFNode     fuselageColor
       field        SFNode     cockpitColor
       field        SFNode     cowlingColor
       field        SFString helicopterName
       eventIn      SFBool inFlight
     ][
       "file:///C:/jakarta-tomcat-
3.2.3/webapps/cocoon/samples/vrml/Savage/AircraftHelicopters/AH1SuperCobra-
UnitedStates/SuperCobraPrototype.wrl#SuperCobra"
       "../../AircraftHelicopters/AH1SuperCobra-UnitedStates/SuperCobraPrototype.wrl#SuperCobra"
     "http://web.nps.navy.mil/~brutzman/Savage/AircraftHelicopters/AH1SuperCobra-
UnitedStates/SuperCobraPrototype.wrl#SuperCobra"
     "file:///C:/jakarta-tomcat-
3.2.3/webapps/cocoon/samples/vrml/Savage/AircraftHelicopters/AH1SuperCobra-
UnitedStates/SuperCobraPrototype.xml#SuperCobra"
     "../../AircraftHelicopters/AH1SuperCobra-UnitedStates/SuperCobraPrototype.xml#SuperCobra"
     "http://web.nps.navy.mil/~brutzman/Savage/AircraftHelicopters/AH1SuperCobra-
UnitedStates/SuperCobraPrototype.xml#SuperCobra"
      ]
     # =================
     NavigationInfo {
       avatarSize [ 2 2 2 ]
       speed 100
       type [ "EXAMINE" "ANY" ]
       visibilityLimit 60000
     }
     Background {
       groundAngle [ 1.309, 1.571 ]
```

```
        groundColor [ 0 0.2 0.4,  0.05 0.25 0.6,  0.1 0.2 0.4 ]
        skyAngle [ 1.309, 1.571 ]
        skyColor [ 0.1 0.1 0.4,  0.1 0.125 0.4,  0.1 0.1 0.3 ]
      }
      Viewpoint {
        description "Amphibious raid - full scenario seen from above"
        orientation -0.557 -0.799 -0.229 0.9506
        position -600 10000 25000
      }
      Viewpoint {
        description "1000m elevation view"
        orientation 0 1 0 -0.78
        position -200 1000 22000
      }
      Transform {
        rotation 0 0 1 .1
        children [
        Viewpoint {
          description "Waypoint"
          orientation 0 1 0 1.57
          position 9451 20 14935
        }
        Viewpoint {
          description "waypoint 2"
          orientation 0 1 0 1.57
          position 11217.2 10.0 13996.6
        }
        ]
      }
      # ViewPositionOrientation is a prototype lets us find good viewpoints for inclusion in the
scene
      ViewPositionOrientation {
        enabled  TRUE
      }
```

```
# This is a hack to orient camera to scene. Actually, scene elements need to be rotated.
Transform {
  rotation 0 1 0 1.57
  children [
  ### XSL-translation warning:  nodeTypeHint unknown for ProtoInstance
CameraCompass36 {
    enabled  TRUE
    positionOffsetFromCamera  0 5 0
    markerColor  0.9 0.9 0.9
    labelColor  0.9 0.9 0.9
  }
  ]
}
DEF CobraCapStartingPoint HiddenViewpoint {
  position  6530 18 16800
  rotation  0 1 0 -0.1
  sensorRadius  100
  label  [ "Cobra CAP" "starting point" "" "6530 11.5 16800" ]
  labelOffset  0 -2 0
  labelFontSize  0.5
  labelColor  1 0.5 0
}
DEF CobraCapInitialClimb HiddenViewpoint {
  position  6530 500 16800
  rotation  0 1 0 -0.4
  sensorRadius  100
  label  [ "Cobra CAP" "initial climb point" "" "6530 500 16800" ]
  labelOffset  0 1 0
  labelFontSize  0.5
  labelColor  1 0.5 0
}
DEF SouthernCapTurnPoint HiddenViewpoint {
  position  25400 500 19100
  rotation  0 1 0 0.4
```

```
        sensorRadius  200
        label  [ "Cobra CAP" "Southern Turn Point" "" "25400 500 191000" ]
        labelOffset  0 1 0
        labelFontSize  0.5
        labelColor  1 0.5 0
      }
      DEF NorthernCapTurnPoint HiddenViewpoint {
        position  5100 500 700
        rotation  0 1 0 -2.3
        sensorRadius  200
        label  [ "Cobra CAP" "Northern Turn Point" "" "5100 500 700" ]
        labelOffset  0 1 0
        labelFontSize  0.5
        labelColor  1 0.5 0
      }
      Switch {
        whichChoice 0
        choice [
        DEF EntireScene Group {
          children [
          Transform {
            children [
            Inline {
              url                           [                          "file:///C:/jakarta-tomcat-
3.2.3/webapps/cocoon/samples/vrml/Savage/Locations/CampPendletonCalifornia/CampPendletonOperatingA
reasExample.wrl"
      "../../Locations/CampPendletonCalifornia/CampPendletonOperatingAreasExample.wrl"
      "http://web.nps.navy.mil/~brutzman/Savage/Locations/CampPendletonCalifornia/CampPendletonOpe
ratingAreasExample.wrl"
      "file:///C:/jakarta-tomcat-
3.2.3/webapps/cocoon/samples/vrml/Savage/Locations/CampPendletonCalifornia/CampPendletonOperatingA
reasExample.xml"
      "../../Locations/CampPendletonCalifornia/CampPendletonOperatingAreasExample.xml"
      "http://web.nps.navy.mil/~brutzman/Savage/Locations/CampPendletonCalifornia/CampPendletonOpe
ratingAreasExample.xml"
       ]
            }
            ]
          }
          DEF LPD Transform {
            translation 6600 10 16800
            children [
            Inline {
              url                          [                           "file:///C:/jakarta-tomcat-
3.2.3/webapps/cocoon/samples/vrml/Savage/Ships/LandingPlatformDock-LPD/LPD.wrl"
      "../../Ships/LandingPlatformDock-LPD/LPD.wrl"
      "http://web.nps.navy.mil/~brutzman/Savage/Ships/LandingPlatformDock-LPD/LPD.wrl"
      "file:///C:/jakarta-tomcat-
3.2.3/webapps/cocoon/samples/vrml/Savage/Ships/LandingPlatformDock-LPD/LPD.xml"
      "../../Ships/LandingPlatformDock-LPD/LPD.xml"
      "http://web.nps.navy.mil/~brutzman/Savage/Ships/LandingPlatformDock-LPD/LPD.xml"
       ]
            }
            ]
          }
          DEF AAAV1_Clock TimeSensor {
            loop TRUE
          }
          DEF AAAV_1_WaypointInterpolator WaypointInterpolator {
            # need to improve elevations on beach (final waypoint)
            # 1 knot = 0.514444444 meters/second
            wayPoints  [ 6802 1 16343

      7685.0 1.0 15873.7,
      8568.1 1.0 15404.4,
      9451.1 1.0 14935.2,
      10334.2 1.0 14465.9,
11217.2 1.0 13996.6,
12100.3 1.0 13527.3,
12983.3 1.0 13058.0,
13866.4 1.0 12588.7,
14749.4 1.0 12119.5,
15632.5 1.0 11650.2,
16515.5 1.0 11180.9,
17398.6 1.0 10711.6,
18281.6 1.0 10242.3,
19164.7 1.0 9773.1,
19400.0 1.0 9648.0,

19782 1 9185
20100 10 8800

 ]
      defaultSpeed  50
      lineColor  0.8 0.8 0
      labelColor  0.8 0.8 0
      labelOffset  0 3 0
      labelFontSize  1.5
      traceEnabled  FALSE
    }
    Transform {
      translation 6600 0 16800
    }
    DEF AAAV-Location EspduTransform {
      readInterval 0
      traceOffset 0 4 0
      traceSize 2 2 2
      translation 6600 0 16800
      writeInterval 1
      children [
      Group {
             children [
             Viewpoint {
               description "AAAV raid from behind"
               orientation 0 1 0 -1.57
               position -40 2 0
             }
             Viewpoint {
               description "AAAV raid from ahead"
               orientation 0 1 0 1.57
               position 50 12 0
             }
             DEF VerticalCorrection Transform {
               translation 0 -4 0
               children [
               DEF AAAV Inline {
                 url                         [                         "file:///C:/jakarta-tomcat-
3.2.3/webapps/cocoon/samples/vrml/Savage/AmphibiousVehicles/AAAV/AAAV.wrl"
     "../../AmphibiousVehicles/AAAV/AAAV.wrl"
     "http://web.nps.navy.mil/~brutzman/Savage/AmphibiousVehicles/AAAV/AAAV.wrl"
     "file:///C:/jakarta-tomcat-
3.2.3/webapps/cocoon/samples/vrml/Savage/AmphibiousVehicles/AAAV/AAAV.xml"
     "../../AmphibiousVehicles/AAAV/AAAV.xml"
     "http://web.nps.navy.mil/~brutzman/Savage/AmphibiousVehicles/AAAV/AAAV.xml"
      ]
               }
               ]
             }
             DEF OtherAAAVs Group {
               children [
               Transform {
                 translation -15 -4 15
                 children [
                 DEF AAAV-2 Transform {
                   children [
              USE AAAV
              DEF AAV-2PATH PositionInterpolator {
                key [ 0.00, 0.11, 0.17, 0.22,
0.33, 0.44, 0.50, 0.55,
0.66, 0.77, 0.83, 0.88,
0.99 ]
                keyValue [ 0.0 0.0  0.0,  1.0 0.96 1.0,
1.5 0.21 1.5,  2.0 0.96 2.0,
3.0 0.0  3.0,  2.5 0.96 3.0,
1.75 0.41 3.0,  1.0 0.96 3.0,
3.0 0.0  3.0,  2.0 0.46 2.0,
1.0 0.4 1.5,  0.0 0.46 1.0,
0.0 0.0  0.0 ]
              }
              ]
            ROUTE AAV-2PATH.value_changed TO AAAV-2.set_translation
            }
            ]
          }
          Transform {
            translation -15 -4 -17.5
            children [
            DEF AAAV-3 Transform {
              children [
              USE AAAV
              DEF AAV-3PATH PositionInterpolator {
                key [ 0.00, 0.07, 0.13, 0.22,
0.36, 0.47, 0.55, 0.625,
0.66, 0.72, 0.80, 0.85,
0.99 ]
                keyValue [ 0.0 0.0  0.0,  1.0 0.96 1.0,
1.5 0.21 2.25,  2.0 0.46 2.66,
3.0 -0.25  3.0,  2.5 0.96 3.0,
1.75 0.41 3.0,  1.0 0.96 3.0,
3.0 0.0  3.0,  2.0 0.46 2.0,
1.0 0.4 1.5,  0.0 0.46 1.0,
0.0 0.0  0.0 ]
              }
              ]
            ROUTE AAV-3PATH.value_changed TO AAAV-3.set_translation
            }
            ]
          }
          ]
        }
        ]
      }
      ]
    }
    DEF Helo-1 Transform {
      translation 6530 11.5 16800
      children [
      DEF HeloInstance1 AH1SuperCobra {
        helicopterName  "Cobra CAP 1"
        tailBoomColor            DEF GREY_APPEARANCE Appearance {
            material Material {
              diffuseColor 0.5 0.5 0.5
            }
          }
        tailRotorDriveShaftColor            USE GREY_APPEARANCE
        horizontalStabilizerColor            Appearance {
            material Material {
              diffuseColor 0.5 0.5 0.5
            }
          }
        fuselageColor            Appearance {
            material Material {
              diffuseColor 0.6 0.6 0.6
            }
          }
        cockpitColor            Appearance {
            material Material {
              diffuseColor 0.6 0.6 0.6
            }
          }
        cowlingColor            Appearance {
            material Material {
              diffuseColor 0.5 0.5 0.5
            }
          }
        inFlight  FALSE
      }
      ]
    }
    DEF Helo_1_WaypointInterpolator WaypointInterpolator {
      # 1 knot = 0.514444444 meters/second
      wayPoints  [ 6600 600 16800
7592.6 600 16921.4
8585.2 600 17042.9
9577.8 600 17164.3
10570.4 600 17285.7
11563.0 600 17407.2
12555.6 600 17528.6
13548.2 600 17650.0
14540.8 600 17771.5
15533.4 600 17892.9

16526.0 600 18014.4
17518.6 600 18135.8
18511.2 600 18257.2
19503.8 600 18378.7
20496.4 600 18500.1
```

```
21489.0 600 18621.5
22481.6 600 18743.0
23474.2 600 18864.4
24466.8 600 18985.8
25400 600 19100

24906.2 600 18230.4
24412.4 600 17360.9
23918.5 600 16491.3
23424.7 600 15621.8
22930.9 600 14752.2
22437.1 600 13882.6
21943.2 600 13013.1
21449.4 600 12143.5
20955.6 600 11273.9
20800 600 11000

20072.7 600 10313.7
19345.4 600 9627.4
18618.1 600 8941.0
17890.8 600 8254.7
17163.5 600 7568.4
16436.2 600 6882.1
15708.9 600 6195.7
14981.6 600 5509.4
14254.3 600 4823.1
13700 600 4300

12777.6 600 3913.9
11855.1 600 3527.7
10932.7 600 3141.6
10010.2 600 2755.4
9087.8 600 2369.3
8165.4 600 1983.2
```

```
7242.9 600 1597.0
6320.5 600 1210.9
5398.0 600 824.8
5100 600 700

5192.8 600 1695.7
5285.5 600 2691.4
5378.3 600 3687.1
5471.1 600 4682.8
5563.8 600 5678.4
5656.6 600 6674.1
5749.4 600 7669.8
5842.1 600 8665.5
5934.9 600 9661.2
6027.7 600 10656.9

6120.4 600 11652.6
6213.2 600 12648.3
6306.0 600 13643.9
6398.7 600 14639.6
6491.5 600 15635.3
6584.3 600 16631.0
6600 600 16800 ]
      defaultSpeed  100
      pitchUpDownForVerticalWayPoints  FALSE
      lineColor  0 0.8 0
      labelColor  0 0.8 0
      labelOffset  0 5 0
      labelFontSize  3
      traceEnabled  FALSE
    }
    DEF Helo1_Clock TimeSensor {
      loop TRUE
    }
```

```
DEF Helo-2 Transform {
  translation 6600 50 16800
  children [
  Transform {
    children [
    DEF HeloInstance2 AH1SuperCobra {
      helicopterName  "Cobra CAP 2"
      tailBoomColor              Appearance {
          material Material {
            diffuseColor 0.5 0.5 0.5
          }
        }
      tailRotorDriveShaftColor              Appearance {
          material Material {
            diffuseColor 0.5 0.5 0.5
          }
        }
      horizontalStabilizerColor              Appearance {
          material Material {
            diffuseColor 0.5 0.5 0.5
          }
        }
      fuselageColor              Appearance {
          material Material {
            diffuseColor 0.6 0.6 0.6
          }
        }
      cockpitColor              Appearance {
          material Material {
            diffuseColor 0.6 0.6 0.6
          }
        }
      cowlingColor              Appearance {
          material Material {
                diffuseColor 0.5 0.5 0.5
              }
            }
          inFlight  TRUE
        }
        ]
      }
      ]
    }
    DEF Helo_2_WaypointInterpolator WaypointInterpolator {
      # 1 knot = 0.514444444 meters/second
      wayPoints  [ 6530 11.5 16800
6530 500 16800
6600 500 16800
7592.6 500 16921.4
8585.2 500 17042.9
9577.8 500 17164.3
10570.4 500 17285.7
11563.0 500 17407.2
12555.6 500 17528.6
13548.2 500 17650.0

14540.8 500 17771.5
15533.4 500 17892.9
16526.0 500 18014.4
17518.6 500 18135.8
18511.2 500 18257.2
19503.8 500 18378.7
20496.4 500 18500.1
21489.0 500 18621.5
22481.6 500 18743.0
23474.2 500 18864.4

24466.8 500 18985.8
```

```
25400 500 19100
24906.2 500 18230.4
24412.4 500 17360.9
23918.5 500 16491.3
23424.7 500 15621.8
22930.9 500 14752.2
22437.1 500 13882.6
21943.2 500 13013.1
21449.4 500 12143.5

20955.6 500 11273.9
20800 500 11000
20072.7 500 10313.7
19345.4 500 9627.4
18618.1 500 8941.0
17890.8 500 8254.7
17163.5 500 7568.4
16436.2 500 6882.1
15708.9 500 6195.7
14981.6 500 5509.4

14254.3 500 4823.1
13700 500 4300
12777.6 500 3913.9
11855.1 500 3527.7
10932.7 500 3141.6
10010.2 500 2755.4
9087.8 500 2369.3
8165.4 500 1983.2
7242.9 500 1597.0
6320.5 500 1210.9

5398.0 500 824.8
5100 500 700
```

```
5192.8 500 1695.7
5285.5 500 2691.4
5378.3 500 3687.1
5471.1 500 4682.8
5563.8 500 5678.4
5656.6 500 6674.1
5749.4 500 7669.8
5842.1 500 8665.5

5934.9 500 9661.2
6027.7 500 10656.9
6120.4 500 11652.6
6213.2 500 12648.3
6306.0 500 13643.9
6398.7 500 14639.6
6491.5 500 15635.3
6584.3 500 16631.0
6600  500  16800 ]
      defaultSpeed  200
      lineColor  0.8 0 0
      labelColor  0.8 0 0
      labelOffset  0 5 0
      labelFontSize  3
      traceEnabled  FALSE
    }
    DEF Helo2_Clock TimeSensor {
      loop TRUE
    }
    DEF AAV7PA1Location Transform {
      scale 1.25 1.25 1.25
      translation 6620 -2 16900
      children [
      Inline {
              url                            [                           "file:///C:/jakarta-tomcat-
3.2.3/webapps/cocoon/samples/vrml/Savage/AmphibiousVehicles/AAV/AAV.wrl"
      "../../AmphibiousVehicles/AAV/AAV.wrl"
      "http://web.nps.navy.mil/~brutzman/Savage/AmphibiousVehicles/AAV/AAV.wrl"
      "                                                                   file:///C:/jakarta-tomcat-
3.2.3/webapps/cocoon/samples/vrml/Savage/AmphibiousVehicles/AAV/AAV.xml"
      "../../AmphibiousVehicles/AAV/AAV.xml"
      "http://web.nps.navy.mil/~brutzman/Savage/AmphibiousVehicles/AAV/AAV.xml" ]
            }
            ]
          }
          DEF AAV7PA1WaypointInterpolator WaypointInterpolator {
            # 1 knot = 0.514444444 meters/second
            wayPoints  [ 6620 -1 16900
      6720 -1 16900
      6870 -1 19500
      7400 -1 19500
      8400 -1 16900
      6620 -1 16900 ]
            defaultSpeed  5
            lineColor  0.8 0 0.65
            labelColor  0.8 0 0
            labelOffset  0 5 0
            labelFontSize  3
            traceEnabled  FALSE
          }
          DEF AAV7PA1Clock TimeSensor {
            loop TRUE
          }
          ]
        ROUTE AAAV1_Clock.fraction_changed TO AAAV_1_WaypointInterpolator.fraction
        ROUTE AAAV_1_WaypointInterpolator.totalDuration TO AAAV1_Clock.cycleInterval
        ROUTE AAAV_1_WaypointInterpolator.positionValue TO AAAV-Location.set_translation
        ROUTE AAAV_1_WaypointInterpolator.orientationValue TO AAAV-Location.set_rotation
        }
        Switch {
          whichChoice 0
          choice [
          Transform {
            rotation 0 1 0 1.5708
            translation 11900 0 6000
            children [
            Inline {
              url [ "file:///C:/jakarta-tomcat-
3.2.3/webapps/cocoon/samples/vrml/Savage/Locations/CampPendletonCalifornia/ChartletTranscribed.wrl
"
      "../../Locations/CampPendletonCalifornia/ChartletTranscribed.wrl"
      "http://web.nps.navy.mil/~brutzman/Savage/Locations/CampPendletonCalifornia/ChartletTranscri
bed.wrl"
      "file:///C:/jakarta-tomcat-
3.2.3/webapps/cocoon/samples/vrml/Savage/Locations/CampPendletonCalifornia/ChartletTranscribed.xml
"
      "../../Locations/CampPendletonCalifornia/ChartletTranscribed.xml"
      "http://web.nps.navy.mil/~brutzman/Savage/Locations/CampPendletonCalifornia/ChartletTranscri
bed.xml"
       ]
            }
            ]
          }
          Group {
            # Southwest corner reference point: 33:12N 117:38W
            # GeoLocation and USE EntireScene to follow
          }
          ]
        }
        ]
      }
      # AAAV-1 Waypoint Routes
# Helo-1 Waypoint Routes
ROUTE Helo1_Clock.isActive TO HeloInstance1.inFlight
ROUTE Helo_1_WaypointInterpolator.totalDuration TO Helo1_Clock.cycleInterval
ROUTE Helo1_Clock.fraction_changed TO Helo_1_WaypointInterpolator.fraction
ROUTE Helo_1_WaypointInterpolator.positionValue TO Helo-1.set_translation
ROUTE Helo_1_WaypointInterpolator.orientationValue TO Helo-1.set_rotation
# Helo-2 Waypoint Routes
ROUTE Helo2_Clock.isActive TO HeloInstance2.inFlight
ROUTE Helo_2_WaypointInterpolator.totalDuration TO Helo2_Clock.cycleInterval
ROUTE Helo2_Clock.fraction_changed TO Helo_2_WaypointInterpolator.fraction
ROUTE Helo_2_WaypointInterpolator.positionValue TO Helo-2.set_translation
ROUTE Helo_2_WaypointInterpolator.orientationValue TO Helo-2.set_rotation
# AAV2
ROUTE AAAV1_Clock.fraction_changed TO AAV-2PATH.set_fraction
# AAV7PA1 Waypoint Routes
ROUTE AAV7PA1WaypointInterpolator.totalDuration TO AAV7PA1Clock.cycleInterval
ROUTE AAV7PA1Clock.fraction_changed TO AAV7PA1WaypointInterpolator.fraction
ROUTE AAV7PA1WaypointInterpolator.positionValue TO AAV7PA1Location.set_translation
ROUTE AAV7PA1WaypointInterpolator.orientationValue TO AAV7PA1Location.set_rotation
<!-- This is the end of the WRL file code -->
 </vrml>
</page>
```

J. X3DCOCOONSELECTION.XHTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- developed with XML Spy v4.0 U (http://www.xmlspy.com) by Clifton Williams (Naval Postgraduate School) -->
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>NPS XML Server Home Page</title>
</head>
<body bgcolor="#FFFFFF" text="#000000" link="#0000FF" vlink=
"#000080" alink="#FF0000">
<p><img border="0" src="images/NPS_r01_c1.gif" width="969"
height="90" align="middle" /></p>

<p align="left"><img border="0" src="images/Flag.gif" align=
"left" width="100" height="55" /></p>

<p align="center"><img border="0" src=
"images/HerrmannHallHomePage.jpg" width="335" height="232" align=
"middle" /></p>

<hr width="50%" size="8" />
<h1 align="center">Cocoon web-publishing framework </h1>

<h2 align="center">X3D Selection Servlet Application</h2>

<hr />
<p>This page is currently set up to support  X3D XML documents.
The X3DCoocon Servlet is called which call the DTD, Xml Parser,
X3D documents and X3D stylesheets use to convert the X3d XML
Documents to VRML, HTML, or XML document (Cocoon Wrapper).</p>

<hr />
<form method="POST" action=
"http://www.xmlking.org/jdbcxml/servlet/jdbc.xml.X3dCocoon">
<blockquote>
<p>Select XML Document.</p>

<select name="xmlInput" size="1">
<option value="HelloWorld.xml" selected>Hello World.xml</option>
<option value="KelpForest.xml">Kelp Forest.xml</option>
<option value="AmphibiousRaid.xml">Amphibious Raid</option>
</select>
```

```
<p>Select Stylesheet.</p>

<select name="xsltInput" size="1">
<option value="X3dHTML.xsl" selected>X3d to HTML</option>
<option value="X3dVRML.xsl">X3d to VRML</option>
<option value="X3dCocoon.xsl">X3d to Cocoon</option>
</select></blockquote>

<input type="submit" value="Submit" /> <input type="reset" value=
"Reset" /></form>

<hr />
<p>The program uses Servlet which can be easily setup to call any
XML document and apply any stylesheets to convert the XML
output</p>


<p><applet code="fphover.class" codebase="./" width="180" height=
"24"><param name="color" value="#000080" />
<param name="hovercolor" value="#0000FF" />
<param name="textcolor" value="#FFFFFF" />
<param name="text" value="Java Source Code Page" />
<param name="effect" value="glow" />
<param name="url" valuetype="ref" value=
"http://www.xmlking.org/xmljavacode/X3dCocoon.html" />
</applet></p>
<p><applet code="fphover.class" codebase="./" width="120" height=
"24"><param name="color" value="#000080" />
<param name="hovercolor" value="#0000FF" />
<param name="textcolor" value="#FFFFFF" />
<param name="text" value="Home Page" />
<param name="effect" value="glow" />
<param name="url" valuetype="ref" value=
"http://www.xmlking.org/index.html" />
```

```
</applet></p>
</body>
</html>
```

K. X3DTOVRML97.XSL

```
<?xml version='1.0'?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"
            xmlns:saxon="http://icl.com/saxon" saxon:trace="no">
<!-- Edited with XML Spy v4.0 U (http://www.xmlspy.com) by Clifton Williams (Naval Postgraduate School) to be used with cocoon and add a wrapper. X3dToVrml97 fto allow X3dCoocon Servlet control the output to XML function and Text output  -->
<!--    XSL namespaces are in transition!  Tools are slow to catch up.
   ***  Edit the topmost stylesheet tag on line 2 of this file to match the xmlns namespace URI for your XSL tool. ***
    W3C:
    Saxon:          <xsl:stylesheet xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
    IBM XSLEditor:   <xsl:stylesheet xmlns:xsl='http://www.w3.org/XSL/Transform/1.0'>
    IE 5:           <xsl:stylesheet xmlns:xsl='http://www.w3.org/TR/WD-xsl'>
    XT:             <xsl:stylesheet xmlns:xsl='http://www.w3.org/XSL/Transform'>-->
<!--
  <head>
  <meta name="filename"     content="X3dToVrml97.xsl" />
  <meta name="author"        content="Don Brutzman" />
  <meta name="revised"       content="22 July 2001" />
  <meta name="description" content="XSL stylesheet to convert X3D files to VRML 97 format, fourth draft matching x3d-compromise.dtd" />
  <meta name="url"               content="http://www.web3D.org/TaskGroups/x3d/translation/X3dToVrml97.xsl" />
  </head>
  Recommended tool:
-  SAXON XML Toolkit (and Instant Saxon) from Michael Kay of ICL, http://users.iclway.co.uk/mhkay/saxon
-  Can also be used with Apache server
-->
<!-- Problems and bugs:
 - Script node content hardwired for javascript to eliminate problematic leading whitespace (which crashes CosmoPlayer)
 - still need to check KeySensor, StringSensor, GeoVRML attribute defaults
```

```
  - combine GeoVRML and H-Anim metadata attributes into a single string
  - ensure Double types get translated properly!
  -->
<xsl:strip-space elements="*" />
<xsl:output method="text" encoding="utf-8" media-type="model/vrml" indent="no" cdata-section-elements="Script"/>
<!-- omit-xml-declaration="yes" -->
<!-- indent handled as a passed parameter since output-tag indent ineffective -->
<!-- ****** root:  start of file ****** -->
<xsl:template match="/">
  <!-- VRML 97 header -->
  <xsl:text>#VRML V2.0 utf8&#10;# X3D-to-VRML-97 XSL translation autogenerated by X3dToVrml97.xsl&#10;#
http://www.web3D.org/TaskGroups/x3d/translation/X3dToVrml97.xsl&#10;&#10;</xsl:text>
  <!-- VRML 200x headers -->
  <xsl:apply-templates select="X3D"/>
  <xsl:apply-templates select="X3D/component"/>
  <xsl:if test="X3D/head">
    <xsl:text>&#10;</xsl:text>
    <xsl:apply-templates select="X3D/head"/>
    <xsl:text>&#10;</xsl:text>
  </xsl:if>
  <!-- VRML 97 header -->
  <!-- Insert GeoVrml EXTERNPROTO declarations, if not otherwise provided -->
  <xsl:if test="//GeoCoordinate and not(//ExternProtoDeclare[@name='GeoCoordinate']) and
not(//ProtoDeclare[@name='GeoCoordinate'])">
    <xsl:text>EXTERNPROTO GeoCoordinate [&#10;</xsl:text>
    <xsl:text>  field  SFNode    geoOrigin    # NULL&#10;</xsl:text>
    <xsl:text>  field  MFString  geoSystem    # [ "GDC" ]&#10;</xsl:text>
    <xsl:text>  field  MFString  point        # []&#10;</xsl:text>
    <xsl:text>] [&#10;</xsl:text>
    <xsl:text>    "C:\Program Files\GeoVRML\1.0\protos\GeoCoordinate.wrl#GeoCoordinate"&#10;</xsl:text>
    <xsl:text>    "file:///C|/Program Files/GeoVRML/1.0/protos/GeoCoordinate.wrl#GeoCoordinate"&#10;</xsl:text>
    <xsl:text>    "urn:web3d:geovrml:1.0/protos/GeoCoordinate.wrl#GeoCoordinate"&#10;</xsl:text>
    <xsl:text>    "http://www.geovrml.org/1.0/protos/GeoCoordinate.wrl#GeoCoordinate"&#10;</xsl:text>
    <xsl:text>  ]&#10;</xsl:text>
```

LIST OF REFERENCES

[ADOBE] Adobe Portable Document Format (PDF) reader version 5.0, Adobe Systems Incorporated http://www.adobe.com/products/acrobat

[Ahttp01] Apache HTTP Server Version 1.3.20 Documentation The Apache Software Foundation, [http://httpd.apache.org/dist/httpd/binaries/win32/], 17 May 2001.

[ANT01] Jakarta Project Ant 1.4 beta 2, The Apache Software Foundation, [http://jakarta.apache.org/ant], 21 January 2001

[Axml01] Cocoon Version 1.8.2 Documentation, The Apache XML Project [http://jakarta.apache.org], and 26 January 2001

[Bird99] Bird, Linda, Goodchild, Andrew, Sue, Hoylen, Tun, Zar Zar, "Introduction to XML Schema", October, 1999. Available at: [http://www.dstc.edu.au/Research/Projects/Titanium/papers/XMLSchema]

[Blais01] Blais, Curtis, Brutzman, Don, Horner, Doug, Nicklaus, Shane, "Web-Based 3D Technology for Education: The *SAVAGE* Project", July 2001.

[Brut01] Brutzman, Don, "DIS-Java-VRML Website", September 2001. Available at: http://www.Web3d.org/WorkingGroups/vrtp/dis-java-vrml/

[Brut00] Brutzman, Don, "DIS-Java-VRML Website", September 2001. Available at: http://www.Web3d.org/TaskGroups/x3d/translation/README.X3D-Edit.htm

[BUR01] Eric Burke, JAVA & XSLT, September 2001, published O'Reilly & Associates, Inc

[Chri01] Christie, B., Young, P., *Integrated Development Environment (IDE) for the Construction of a Federation Interoperability Object Model (FIOM)*, Master's Thesis, Naval Postgraduate School, Monterey, CA, September 2001.

[CSJSP] Hall, Marty, *Core Servlets and Java Server Pages*, Sun Microsystems Press, March 2001.

[DII98] U.S. Defense Information Systems Agency, Defense Information Infrastructure Master Plan, Version 7.0, Appendix A, Communications and Computer Infrastructure, [http://www.disa.mil/diimp/diimp-a.html]. 13 March 1998.

[DII01] "DII COE Data Emporium." [http://diides.ncr.disa.mil/xmlreg/user/index.cfm]

[Deitel] Deitel, H.,Deitel, P., *Java How to Program 3rd Edition*, Prentice Hall, 1999.

[DL99] United States Department of Defense, Defense Link, News Release, NO.032-99, Department of Defense Budget For FY 2000, [http://www.defenselink.mil/news/Feb1999/b02011999_bt032-99.html], 1 February 1999.

[DoD91] DoD Directive 8320.1, DoD Data Administration, [http://Web7.whs.osd.mil/pdf/d83201p.pdf], September 26, 1991.

[DWS01] C. S. Yang and M. Y. Luo - National Sun Yat-Sen University, Taiwan, R.O.C "Design and Implementation of an Administration System for Distributed Web Server" June 8, 2001

[Gold01] Goldfarb, Charles F., Prescod, Paul, *The XML Handbook,* Third Edition, Prentice Hall PTR, 2001.

[Hun00] Hunter, D., and others, *Beginning XML*, Wrox Press Ltd, August 2000.

[JSR31] Java Specification Request 31, "XML Data Binding Specification", [http://jcp.org/jsr/detail/031.jsp]. August 1999.

[JV20] Chairman Joint Chiefs of Staff, "Joint Vision 2020", June 2000.

[JVM01] Java Virtual Machine (JVM) in the Java 2 Software Development Kit Package J2SDK-1_3_1_01 http://java.sun.com/j2se/1.3/install-windows.html, Sun Microsystems, Inc

[Leff00] Leffingwell, D., Widrig, D., *Managing Software Requirements A Unified Approach*, Addison-Wesley, 2000.

[KA95] Khoshafian, S., Abnous, R., *Object Orientation*, John Wiley and Sons, Inc., New York, NY, 1995.

[Kay00] Kay, M., *XSLT Programmer's Reference*, Wrox Press Ltd, July 2000.

[KM98] Kahng, J., McLeod D., "Dynamic Classificational Ontologies: Mediation of Information Sharing in Cooperative Federated Database Systems", Cooperative Information Systems, Trends and Directions, Academic Press, 1998.

[Lytt00] Lyttle, B., Ehrhardt, T., *Interconnectivity via a Consolidated Type Hierarchy and XML*, Master's Thesis, Naval Postgraduate School, Monterey, CA, 2000.

[MCL01] Brett Mclaughlin ,JAVA & XML, Second Edition, August 2001, published O'Reilly & Associates, Inc

[MSXML] Microsoft XML Parser (MSXML) version 4.0 http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnmsxml/html/xmlparser.asp , Microsoft Corporation, November 2000

[Nick01] Nicklaus, Shane, *Scenario Authoring And Visualization For Advanced Graphical Environments (*Savage), Master's Thesis, Naval Postgraduate School, Monterey, California, USA, June 2001.

[OECD01] Organization for Economic Co-operation and Development, *Gross Domestic Product Table*, [http://www.oecd.org/std/gdp.htm], Jul 2001.

[Pit97] Pitoura, E., "Providing Database Inter-operability through Object-Oriented Language Constructs", Journal of Systems Integration, Volume 7, No. 2, August 1997, pp. 99-126.

[SAX] Saxon XSLT Processor , Michael Kay http://saxon.sourceforge.net/

[SB01] SeeBeyond Technology Corp., *SeeBeyond Taking the Challenges out of eBusiness Integration, Making eBusiness a Reality*, Marketing Overview, [http://www.seebeyond.com/aboutus/], Jul 2001.

[Sun01] Reinhold, M., *The Java™ Architecture for XML Binding (JAXB), Working-Draft Specification*, Sun Microsystems, Inc., [ftp://ftp.java.sun.com/pub/xml/987dfjaxb10ea3ds/jaxb-0_21-wd-spec.pdf]. 30 May 2001.

[Tom01] Jakarta-Tomcat Version 3.2.3 beta 4 Documentation, The Apache Software Foundation, http://jakarta.apache.org/downloads. 17 July 2001.

[USMTF] Defense Information Systems Agency (DISA), "United States Message Text Formatting (USMTF) Program Home Page", July 2001. Available at: http://www-usmtf.itsi.disa.mil

[VRML] Ames, A. L., Nadeau, D. R., Moreland, J. L., *VRML 2.0 Sourcebook 2nd Edition*, John Wiley & Sons, 1997.

[W3C] World Wide Web Consortium, *Web site*, [http://www.w3c.org].

[W3Csch] World Wide Web Consortium, *XML Schema Web site*, [http://www.w3c.org/XML/Schema].

[WCS00] Walsh, A., Couch, J., Steinberg, D., *Java 2 Bible*, IDG Books Worldwide, Inc., Foster City, CA, 2000.

[WEN01] Task Force Web, "Web Enabling the Navy Overview", August 2001. Available at: [http://quickplace.hq.navy.mil/navyxml

[Whatis] Whatis?com Web site, Searched Engine for a Tech Definitons site. http://searchwebservices.techtarget.com/sDefinition/0,,sid26_gci213404,00.html Last updated on: Nov 24, 2001

[Wie93] Wiederhold, G., "Intelligent Integration of Information", ACM-SIGMOD 93, Washington, DC, May 1993, pp. 434-437.

[Will00] Williams,K.,Brundage, M, Dengler, P., Gabriel, J., Hoskinson, A., Kay, M., Maxwell,T., Ochoa,m., Papa, J., Vanmane, M., *Professional XML Databases*,Wrox Press Ltd. 2000.

[X3D] VRML/X3D Conformance examples at http://www.web3d.org/TaskGroups/x3d/translation/examples/Conformance/contents.html

[XmlSof01] XMLSoftware [http://xmlsoftware.com/publishing] Nov 9,2001.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Road, Suite 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Road
   Monterey, CA 93943-5101

3. Dan Dolk, Code IS 1
   Naval Postgraduate School
   Monterey, CA 93940-5000

4. LCDR Raymond Buettner, Code IW 1
   Naval Postgraduate School
   Monterey, CA 93940-5000

5. Mr. Arison..1
   JCS/J6
   Pentagon
   Washington, D.C. 20350-2000

6. Dr. Michael P. Bailey...1
   Technical Director, Marine Corps Training and Education Command
   Commanding General
   Marine Corps Combat Development Command, Code 46T
   3300 Russell Road
   Quantico, VA 22134

7. Dr. Philip S. Barry 1
   Chief, S&T Initiatives Division
   Defense Modeling and Simulation Office
   1901 N. Beauregard Street, Suite 500
   Alexandria VA 22311

8. Robert J Barton III 1
   Fraunhofer Center for Research in Computer Graphics (CRCG)
   321 South Main St
   Providence, RI 02903

9. Curtis Blais..1
Institute for Joint Warfare Analysis
Naval Postgraduate School
Monterey, CA 93940-5000

10. Gordon Bradley...1
Naval Postgraduate School
Monterey, CA 93940-5000

11. Don Brutzman, Code UW/Br...10
Naval Postgraduate School
Monterey, CA 93940-5000

12. Dan Boger, Code C3/Bo 1
Naval Postgraduate School
Monterey, CA 93940-5000

13. Rex Buddenberg Code IS/Bu...1
Naval Postgraduate School
Monterey, CA 93940-5000

14. Fred Burkley 1
NAVSEA Undersea Warfare Center
Division Newport
Code 2231, Bldg 1171-3
1176 Howell Street
Newport, RI 02841-1708\

15. Bob Cabanya 1
Information Operations, Inc.
1298 Bay Dale Dr.
Arnold, MD 21012

16. LTC Neil Cadwallader 1
MCTSSA, Box 555171
Camp Pendelton, CA 92055-5171

17. Erik Chaum 1
NAVSEA Undersea Warfare Center
Division Newport
Code 2231, Building 1171-3
1176 Howell Street
Newport, RI 02841-1708

18. Rober Clover 1
Institute for Defense Analyses
1801 N. Beauregard St.
Alexandria, VA 22311-1772

19. Colonel William Crain, USA 1
Defense Modeling and Simulation Office
1901 N. Beauregard St. Suite 500
Alexandria, VA 22311

20. Justin Couch and Alan Hudson 1
Yumatech, Inc
600 Malden Ave. East
Suite 202
Seattle, WA 98112

21. Dr. Paul Fishwick 1
Computer & Information Science and Engineering Department
University of Florida
Post Office Box 115120
322 Building CSE
Gainsville, FL 32611-6120

22. Dr. Tony Healey, Code ME/Hy 1
Naval Postgraduate School
Monterey, CA 93943-5101

23. Captain Mike Hunsberger, USAF 1
Air Force Communications Agency/TCPD
203 Lossey St
Scott AFB, IL 62225

24. Pamela Krause 1
Advance Systems & Technology
National Reconnaissance Office
14675 Lee Rd
Chantilly, VA 20151-1714

25. S. David Kwak 1
The Mitre Corporation – M/S B155
202 Burlington Rd.
Bedford, MA 01730-1420

26. John Lademan 1
Electronic Sensors and Systems Sector
Northrop Grumman Corporation
PO Box 1488 – MS 9030
Anapolis, MD 21404

27. Major Dave Laflam, USA...1
Army Model and Simulation Office
Office of the Deputy Chief of Staff for Operations and Plans
1111 Jefferson Davis Highway
Crystal Gateway North (Suite 503E)
Arlington, VA 22202

28. Dr. Francisco Loaiza and Dr. Eugene Simaitis...1
Institute for Defense Analyses
Systems Evaluation Division
1801 N. Beauregard St.
Alexandria, VA 22311

29. Dr. R Bown Loftin 1
Director of Simulation Programs
Virginia Modeling Analysis & Simulation Center
Old Dominion University
7000 College Dr
Suffolk, VA 23435

30. Dell Lunceford...1
Director, Army Model & Simulation Office
Crystal Gateway North Suite 503E
1111 Jefferson Davis Highway
Arlington, VA 22202

31. Mike Macedonia 1
Chief Scientist and Technical Director
US Army STRICOM
12350 Research Parkway
Orlando, FL 32826-3276

32. Fahrid Mamaghani...1
19223 SE 45th St
Issaquah, WA 98027

33. Capt. Maslowsky..1
CNO/N62
Pentagon
Washington, D.C. 20350-2000

34. Michael McCann..1
Monterey Bay Aquarium Research Institute (MBARI)
PO Box 628
Moss Landing, CA 95039-0628

35. Chuck Mirabile 1
USMC Program Office, D12
52560 Hull St.
San Diego, CA
92152-5001

36. Capt Mark Murray USAF 1
Joint Battlespace Infosphere (JBI)
AFRL/IFSE
Building 3, Room E-1078
525 Brooks Road
Rome, NY 13441-4505

37. Michael Myjak..1
Vice President and CTO
The Virtual Workshop
PO Box 98
Titusville, FL 32781

38. Neal Park, President...1
Nexternet, Inc.
2900 Gordon Ave.
Suite 202
Santa Clara, CA 95051

39. Marty Paulsen 1
Analytic Graphics, Inc.
3760 Killroy Airport Way
Suite 270
Long Beach, CA 90806

40. George Philips..1
CNO, N6M1
2000 Navy Pentagon
Room 4C445
Washington, DC 20350-2000

41. David Pratt 1
SAIC

42. Dr. Mark Pullen & Dr. Robert Simon..2
Department of Computer Science/C3I Center MS4A5
George Mason University
FairFax, VA 22030

43. Dick Puk...1
President
Intelligraphicas Incorporated
7644 Cortina Court
Carlsbad, CA 92009-8206

44. CAPT Jason Quigley USAF 1
Joint Battlespace Infosphere (JBI)
AFRL/IFSE
Building 3, Room E-1078
525 Brooks Road
Rome, NY 13441-4505

45. Dr. Martin Reddy 1
SRI International, EK219
333 Ravenswood Avenue
Menlo Park, CA 94025

46. Dr. R. Jay Roland, President...1
Rolands and Associates
500 Sloat Avenue
Monterey CA 93940

47. RADM Paul Sullivan, USN 1
Director, Submarine Warfare Division N77
2000 Navy Pentagon, 4D542
Washington, DC 20350-2000

48. Craig Swanson 1
Science Applications International Corporation
Information Systems Division
1710 SAIC Dr.
Mclean, VA 22102

49. CAPT Rober Voigt, USN 1
Chair, Electrical Engineering Department
U.S. Naval Academy
Annapolis, MD 21402

50. Joe Williams..1
3421 Bonita Vista Lane
Santa Rosa, CA 95404

51. LtCol. Ziegenfuss..1
HQMC C4I Plans and Policy Division
2 Navy Annex
Washington, D.C. 20380-1775

52. Walter H. Zimmers 1
Defense Threat Reduction Agency
CPOC
6801 Telegraph Road
Alexandria VA 22310-3398

53. Dr. Michael Zyda, CodeCS/Zk 1
Director, Modeling Virtual Environments and Simulation (MOVES) Institute
Computer Science Department
Naval Postgraduate School
Monterey, CA 93940-5000

55. Lieutenant Clifton J. Williams 1
1717 Abadie Ave
Metairie, LA 70003